Passive Link Quality Estimation for Accurate and Stable Parent Selection in Dense 6TiSCH Networks

Rodrigo Teles Hermeto, Antoine Gallais
ICube Lab
CNRS / University of Strasbourg
{teleshermeto,gallais}@unistra.fr

Kristof Van Laerhoven
Ubiquitous Computing group
University of Siegen
kvl@eti.uni-siegen.de

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

Abstract

Industrial applications are increasingly demanding more low-power operations, deterministic communications and end-to-end reliability that approaches 100%. By keeping nodes time-synchronized and by employing a channel hopping approach, IEEE 802.15.4-TSCH (Time-Slotted Channel Hoping) aims at providing high-level network reliability. For this, however, we need to construct an accurate schedule, able to exploit reliable paths. In particular, radio links with high Packet Error Rate should not be exploited since they are less energy-efficient (more retransmissions are required) and they negatively impact the reliability. In this work, we take advantage of the continuously advertisement packets transmitted by the nodes to identify neighbors with a good link quality. We argue that when a node ranks its neighbors through their rate of broadcast packets received, it can identify stable parents, even when the data packets use different, collision-free transmission opportunities. Our experiments on a large-scale platform highlight that our approach improves the convergence delay, identifying the best routes to the border router during the bootstrapping (or reconverging) phase without adding any extra control packet.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Distributed Systems; J.4 [**Computer Applications**]: Physical Sciences and Engineering

General Terms

Experimentation, Measurement, Design

Keywords

IoT, 6TiSCH, IEEE802.15.4-2015-TSCH, Link Quality Estimation

International Conference on Embedded Wireless Systems and Networks (EWSN) 2018
14–16 February, Madrid, Spain
© 2018 Copyright is held by the authors.
Permission is granted for indexing in the ACM Digital Library
ISBN: 978-0-9949886-2-1

1 Introduction

Industry 4.0 is currently an emerging approach, aiming to re-use the IoT concepts in the automation world. The so-called Industrial Internet of Things (IIoT) relies on wireless technologies that are able to provide a high Quality of Service (QoS) for a plethora of industrial applications with high requirements concerning the latency and the network reliability [1].

To provide Quality of Service (QoS) for industrial-like wireless networks, IEEE 802.15.4-2015 standard was published in 2016 [2], defining in particular the Time-Slotted Channel Hoping (TSCH) mode. TSCH targets specifically the low-power, deterministic and reliable wireless industrial networks. This standard schedules the transmissions such that each application has enough transmission opportunities while avoiding collisions.

To construct an accurate schedule, the network needs to select the best route(s) for each flow, and to allocate enough transmission opportunities to each node along the path to the border router. While unicast transmissions may be protected against collisions in *dedicated cells*, the control traffic (possibly in broadcast) uses the *shared* cells, transmitted with a best-effort strategy.

Estimating the link quality is of prime interest: selecting a suboptimal preferred parent implies that many packets have to be retransmitted to be correctly received by the next hop. The routing topology is in other words inefficient and the incriminated nodes may quickly run out of energy. Righetti *et al.* highlighted the need of an accurate link estimation before constructing the 6TiSCH schedule: A mis-estimation negatively impacts the RPL convergence [3].

Unfortunately, estimating the link quality isn't trivial. Many proposed approaches rely on active probing to evaluate the link quality toward all the neighbors. While the probing period can be adapted dynamically to reduce the overhead [4], control packets are still required to quickly react to changes.

Worse, active probing is very sensitive in 6TiSCH. Using shared cells for the probes leads to many collisions, even with L2 and L3 control packets. Thus, some propositions rely on reserving a dedicated cell for *each* neighbor to send the probes (e.g. [3]). Such strategy tends to be expensive and suboptimal, though: could we reach the same objective without relying on additional control packets?

In this work, we propose a purely passive approach for 6TiSCH, where a node identifies the best possible parents without testing individually every unicast link. To select the most accurate preferred parent, a node *ranks* its neighbors by the amount of advertisement packets received from each of them and chooses one of the top ranked ones to forward its packets.

To the best of our knowledge, this work is the first to argue that a ranking of the candidates is sufficient: a node does not need to estimate *'exactly'* the link quality through a neighbor. We verify experimentally that this metric may constitute a good estimator even if the broadcast packets are subject to collisions.

The contributions presented in this paper are as follows:

1. We propose a passive approach where a node can identify good neighbors without unicast probes. More precisely, we *rank* different possible parents using only the broadcast rate instead of estimating expensively the Packet Delivery Ratio (PDR). This metric can be used for both bootstrapping and the DODAG reconfiguration;
2. We analyze statistically the size of the observation window in order to minimize the reactivity time while identifying accurately the best candidates parents;
3. We integrate our passive ranking method in the 6TiSCH stack, to achieve an higher stability by avoiding *'blindly'* preferred parent changes;
4. We perform an experimental evaluation on the FIT IoT-LAB to confirm our hypothesis regarding the correlation of broadcast and acknowledged unicast receptions rate. We highlight that a node is able to safely identify one of its best parents, and to construct close to optimal routes.

2 Background and Related Work

We present here the main behavior of the 6TiSCH/IEEE 802.15.4-TSCH stack, to understand more finely the need and the challenge for an accurate link quality estimation. We will then detail the existing work, and why these propositions cannot solve efficiently the link quality estimation in 6TiSCH.

2.1 IEEE 802.15.4-TSCH Overview

Using a different physical channel for successive transmissions reduces the impact of external interference and improves the network reliability [5]. Indeed, the standardization bodies have proposed to use channel-hopping techniques, which allow subsequent packets to be transmitted over different frequencies, mainly to be utilized for industrial wireless networks.

IEEE 802.15.4-2015 has proposed the TSCH mode, largely inspired from the previous ISA100.11a [6] and WirelessHART [7] standards. The slotframe consists of a matrix of *cells* of equal length, each cell being defined by a pair of *timeslot* and *channel* offsets (cf. Figure 1).

The transmissions are organized within this slotframe. The standard defines two different types of cells:

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 *full* access;

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

At each timeslot, a node may transmit or receive a frame, or it may turn to sleep mode for saving energy. A set of timeslots constructs a slotframe. Each timeslot is labeled with an Absolute Sequence Number (ASN), a variable which counts the number of timeslots elapsed since the network was established.

At the beginning of each timeslot, the channel offset is translated into a physical channel using the ASN value:

$$frequency = F\left[\left(ASN + channelOffset\right) \,\%\, nFreq\right] \quad (1)$$

where ASN denotes the Absolute Sequence Number of the timeslot, $channelOffset$ the channel offset of the current cell, $nFreq$ is the number of available channels*, and $F[]$ is a bijective function relying an integer comprised between 0 and $nFreq$ and a physical channel [8]. Finally, note that each cell can be either dedicated (contention-free) or shared (contention-based approach).

Many distributed and centralized scheduling algorithms have been proposed so far for TSCH networks [9]. Whatever the scheduling algorithm, we need to select the most reliable links to forward the packets. The centralized scheduling algorithms assume the link quality of each radio link is known a priori while each mote reserves reactively as many cells as required with its preferred parent in the distributed approach. In both approaches, identifying the best parents is of utmost importance.

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 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 bind IPv6 (i.e. 6LoWPAN) to a reservation based MAC layer (i.e. TSCH). 6TiSCH makes a distinction between the protocol which defines how to negotiate the cells (i.e. 6P [10]) and the algorithm deciding how many cells to allocate in the schedule (the Scheduling Function such as

*16 channels are at most available when using IEEE 802.15.4-compliant radios at 2.4 *GHz*

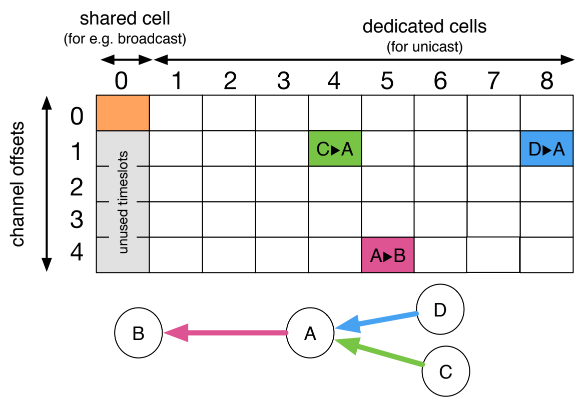

Figure 1. An example of TSCH scheduling for node D. A → D stands for *'A transmits to D'*, while the shared cell is used for broadcast or control frames.

SF0 [11]). The solution is very flexible since any scheduling algorithm may be practically implemented: a new Scheduling Function has just to be defined and interfaced with 6P. Typically, 6P packets use the shared cells since they must be exchanged with some radio neighbors with which dedicated cells are yet to be negotiated.

6TiSCH minimal [12] advocates the usage of one at the beginning of the slotframe, used for control packets, possibly in broadcast. For instance, Enhanced Beacons (EBs) are transmitted during the shared timeslot so that new nodes may associate with the existing network. The rest of the slotframe comprises dedicated cells, which are reserved through 6P.

2.3 Link Quality Estimation

While the radio links exhibit very different link qualities in wireless networks, the routing protocol should select the shortest but most reliable paths [13]. Optimizing the service provided by the link layer to the network layer thus imposes to minimize the link layer transmissions required for a data packet to be correctly delivered (and acknowledged) by the sink station.

Dawans *et al.* already demonstrated that using a default link value tends to privilege the selection of newly discovered neighbors as preferred parents [14]. Thus, updating the link quality continuously is not enough: the quality toward *all* the neighbors has to be more or less roughly estimated.

2.3.1 Active Monitoring

A node may estimate the link quality *actively* toward a neighbor, by transmitting probe packets. MoMoRo proposes to combine several metrics (ETX, RSSI and LQI) to estimate the link quality [15]. Bildea *et al.* propose to categorize the different links, discriminating good, bad and intermediate link qualities with a Gilbert-Elliot model [16]. We are also convinced that an exact link quality is not mandatory to select one of the best radio links to route the packets.

The last version of Contiki (3.0) implements a probing method: each inactive neighbor is probed periodically to discover *better* next hops. Vallati *et al.* [17] verified experimentally that probes allow the nodes to select only the best links, and to reduce globally the packet losses. Unfortunately, probes are sources of energy wastage and create collisions with other control packets. Besides, sending probes independently to each possible neighbor is too expensive for dense topologies.

Pradittasnee *et al.* proposed to use broadcast probes to reduce the overhead [18]: each node knows the amount of received probes to infer the packet delivery ratio. Even though broadcast probes induce lower costs (e.g. energy, processing) than unicasted ones, they still increase the probabilities of collisions with other control packets and resulting expensive retransmissions.

Consequently, we propose rather to avoid such probes, by using a passive estimation of the link quality, based on the monitoring of existing control packets.

2.3.2 Passive Monitoring

Alternatively, passive approaches only use the existing traffic to infer the link quality. The monitoring process does not disturb the network (i.e. collisions and energy consumption).

Gomes *et al.* propose to introduce LQE nodes dedicated to traffic monitoring and link quality estimation [19]. Typically, LQE nodes are all the nodes which have to forward traffic. They use the RSSI and LQI values to infer the Packet Delivery Ratio (PDR) of the given links. Unfortunately, RSSI and PRR have been proved to be only loosely correlated in many situations [20].

LPL approaches use a preamble before each data transmission so that the receiver stays awake to receive the packet. Overhearing may be used to affine the link quality estimation, even if the node does not exchange packets with one of its neighbors [21]. In that case, a node must capture all the packets even if it is not the link layer destination. Then, it uses the RSSI of the received frame after having properly inferred the identifier of the transmitter to estimate the ratio of corrupted frames.

However, data packets are exchanged only with **active** neighbors. Thus, the link quality toward an inactive neighbor can only be estimated if the node overhears the traffic. Unfortunately, overhearing is expensive, and may even be impossible in multichannel (this neighbor may use a different channel offset during the same timeslot).

2.3.3 Hybrid Monitoring

Hybrid approaches try to combine both active (active neighbor) and passive (inactive neighbor) methods. Thus, probing packets are used only when no data traffic is available.

For instance, 4-Bit Link estimation mixes active (probe packets) and passive (ETX for data packets) criteria [22] so that probes are only used when no data packet is exchanged with a given neighbor. Probes are broadcast to mutualize the overhead: one single probe helps to refine the link quality toward *all* the neighbors.

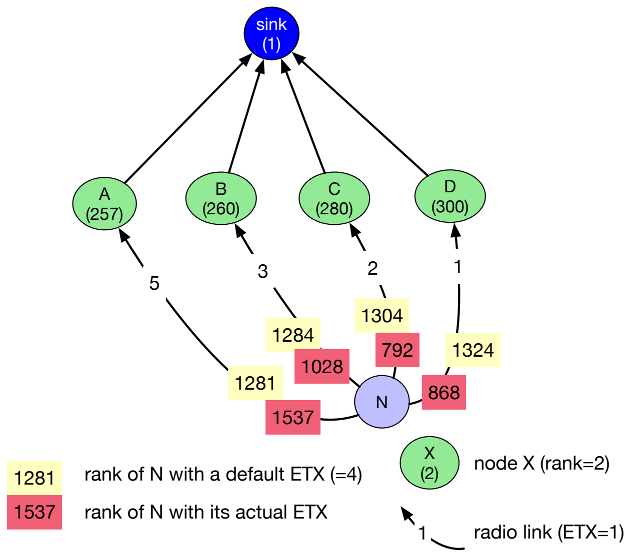

Figure 2. Convergence of RPL when using a initial default link metric.

3 Problem Statement

We need to identify good links to construct efficient routes. In particular, a node N has to *estimate* the link quality toward:

- its **active neighbors**, with which it exchanges data packets. Typically, an active neighbor is a node for which it forwards the packets, or to which it sends its own traffic, a next top toward the border router. A passive method is easy to implement: it is sufficient to measure the Packet Delivery Ratio.
- its **inactive neighbors**, with which no unicast packet is exchanged. In 6TiSCH, no dedicated cell is reserved to these neighbors, and no data packet is available for a passive measurement. Nevertheless, an inactive neighbor may be selected as preferred parent when the radio link qualities change or if the primary parent crashes.

3.1 Limits of existing approaches

We assume that RPL is used for routing, and we use the objective function OF0 to compute the rank, based on the ETX of the links[†]. 6TiSCH minimal [12] does not recommend any default ETX value for inactive neighbors. The OpenWSN implementation[‡] uses a default link cost of 4, i.e. the ETX toward an inactive neighbor is assumed to be 4.

A node N computes its own rank from those of its preferred parent P:

$$rank(N) = rank(P) + MinHopRankIncrease * ETX_{N \to P} \tag{2}$$

with $rank(S)$ denoting the rank of the node S, `MinHopRankIncrease` is a constant (by default equal to 256), and $ETX_{N \to P}$ denotes here the ETX value from N to P.

Let us consider the topology depicted in Figure 2. Since all the neighbors are considered initially as *inactive*, all the ETX values are assumed to be equal. Thus, N picks as preferred parent its neighbor with the lowest rank. After its association, N sends some packets to A and is able to refine its ETX estimation to reflect the actual link quality. Thus, its rank is updated from 1281 ($= rank(A) + 256 * 4$) to 1537 ($= rank(A) + 256 * 5$) N detects that another neighbor (B) would provide a lower rank: N will change its preferred parent to B. If the actual ETX value is superior than the default ETX, N will probe iteratively each neighbor.

Inversely, using a too large default ETX is also suboptimal. In Figure 2, N will not select C or D as preferred parents although they provide a better path to the border router. Indeed, their rank with the default value would be 1304 and 1324 respectively. Since their default ETX is too large, these nodes will never be selected, without any opportunity to re-estimate more accurately the link quality.

The problem becomes even trickier to handle with temporal variations, very common for this kind of scenario.For inactive neighbors, the link metric was evaluated a long time ago and does not reflect the current quality. De facto, these neighbors will never be considered again to serve as preferred parent, except if the current one crashes or its link quality becomes very bad (i.e. ETX > 4).

For higher network density, a node may limit the number of neighbors to be included in its neighbors table. In such scenario, the nodes exclude periodically from their neighborhood table their worst neighbors. Later, these neighbors might be added back with the default link cost until they are probed again. With this inclusion/removal, a node may consider again a bad neighbor when a parent changing is required.

3.2 Challenges

A naive approach would consist in probing each radio link individually to measure the Packet Error Rate. However, such an approach is practically inapplicable:

- each neighbor must be probed independently, generating a large volume of control packets. Such an estimation phase would waste too much energy as it should be executed continuously to detect radio link quality changes;
- if the probes use the shared cells, they may collide with other sensitive packets such as EB and routing packets. Possibly, such method would disturb the routing and synchronization protocols, thus preventing the network from converging properly.

We have to tackle the following challenges:

1. **Inactivity**: the link toward all the nodes must be estimated. In a 6TiSCH network, a node only communicates with its preferred parent and its RPL children [24], while its others neighbors are inactive;
2. **Passiveness**: the link quality should be estimated passively to reduce the energy consumption;
3. **Reactivity**: the estimation must handle link quality variations, and should be able to recover when the link quality toward a neighbor significantly decreases;
4. **Stability**: the estimator should be tolerant to short-terms variations to avoid excessive parents switching;

[†] We use here the default parameters for OF0 [23]

[‡] http://www.openwsn.org/

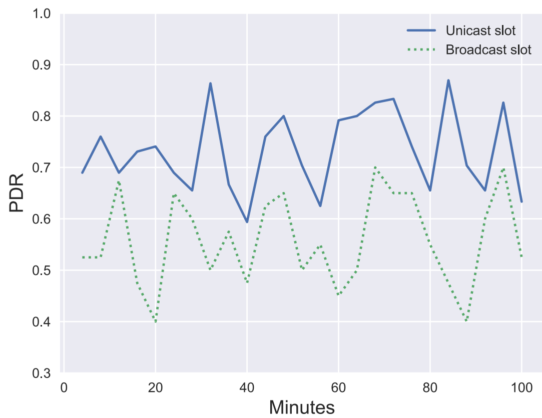

Figure 3. Difference between the PDR of unicast and broadcast slots between two nodes.

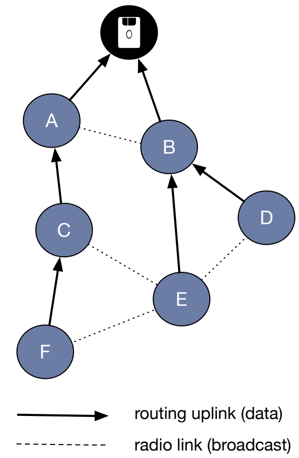

Figure 4. A multi-hop network with its uplinks and downlinks.

5. **Multichannel**: we have to accurately estimate the link quality even when multiple channels are used. In particular, the different channels may exhibit a very different PDR [25], thus distorting the estimation when only a few packets are used to compute the PDR;

6. **Broadcast accuracy**: broadcast packets use shared cells (with collisions) while data packets use dedicated (contention free) cells. Thus, the PDR of both types of cells is very different and transforming the broadcast PDR into an unicast PDR cannot be made easily.

   For instance, Figure 3 illustrates the evolution of such different PDR, as obtained in shared and dedicated cells with a star network of 11 nodes (experimental setup described in Section 4.1).

3.3 Scenario & Assumptions

We focus here on a convergecast multihop network, since bidirectional traffic is still not efficiently supported by 6TiSCH and RPL [26]. Figure 4 illustrates a typical topology: the plain lines represent the uplinks selected by the routing protocol (RPL in our performance evaluation). Each node selects a preferred parent, to which it sends all its unicast traffic.

To be received correctly, the Enhanced Beacons (EB) and the RPL's DODAG Information Object (DIO) have to be transmitted during the shared cells, when all the nodes have to stay awake. Thus, a node continuously receives broadcast packets from its neighbors. Since the broadcast packets are never acknowledged, no backoff is used before the transmission, and the control packets may collide. Figure 4 depicts the outgoing transmissions of node E. The broadcast packets are received by B,D, C and F, while unicast packets are received by B only.

Our hypothesis is that there is a correlation between the reception rate of advertisement packets from a given neighbor and the link quality it would provide as a preferred parent. In other words, if the broadcast rate of $C \rightarrow F$ is larger than those of $E \rightarrow F$, the unicast PDR is assumed to be higher for the radio link $F \rightarrow C$ than for the link $F \rightarrow E$.

As mentioned in Section 2.1, a TSCH slotframe is composed of shared and dedicated slots. They are used in our scenario in the following way:

1. shared cells are used to transmit EB or DIO packets (broadcast);
2. dedicated cells are only used to forward the data and DAO packets (unicast toward the border router).

Can we use the **broadcast packets sent through shared cells** prone to collisions to **classify** the link quality of the **unicast collision free cells**?

4 Can we use the broadcast rate to estimate the unicast, contention-free link quality?

Thus, a broadcast packet may not be received correctly because of:

collisions: broadcast packets have to be transmitted during the shared cells, i.e. with contention, so that all the neighbors are awake to receive them;

SNR: a bad link quality implies some packets are not received by some of the neighbors (statistically with the worst SNR).

However, unicast packets are protected in dedicated cells and do suffer only from the physical PER. We investigate here how the broadcast rate can allow to identify good links to use for unicast transmissions.

4.1 FIT Iot-LAB & Hardware

In this Section, we present our experimental campaign over the FIT IoT-LAB platform, an open large-scale and multiuser infrastructure. The FIT IoT-LAB platform is a shared platform with potential concurrent experiments, thus providing a realistic environment for IoT-related systems and applications experiments.

We employ the A8-M3 motes, based on ARM3505 (ARM Cortex A8) combined with a STM32 micro-controller and a AT86RF231 radio chipset, providing an IEEE 802.15.4 compliant PHY layer. We also execute OpenWSN that implements the full 6TiSCH stack (i.e. IEEE 802.15.4-TSCH, 6P, SF0, 6LoWPAN and RPL).

We implemented our proposal in OpenWSN, where nodes register the number of broadcast packets sent by their neighbors. To control more finely the experiments, and to mimic radio links with different qualities, we pre-installed a schedule at the compilation time (without collisions for dedicated cells). Our modifications are freely available on GitHub [§].

4.2 Measurement

We use here the topology depicted in Figure 5. We focus on a specific node (hereafter designated as *target node*) which sends its packets periodically to each neighbor at a time. The radio link quality depends coarsely on the distance to the target node.

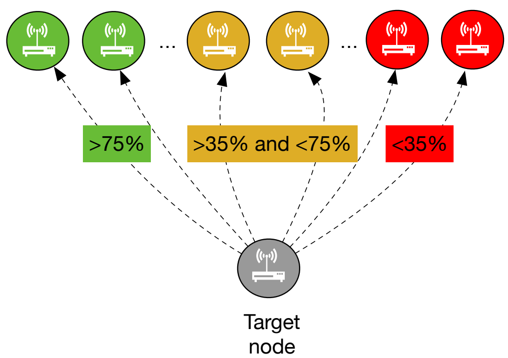

Figure 5. Topology with different links qualities.

We compute in a centralized manner a schedule so that:

1. the shared cells are used to transmit the enhanced beacons (EB) and the RPL's DIOs. The target node tracks all the broadcast packets received from each of its neighbors;
2. a dedicated cell is reserved with each neighbor. This cell is used to transmit unicast (data) packets to compute the unicast Packet Delivery Ratio (PDR). The target node would expect to obtain this PDR if it chooses the corresponding neighbor as next hop toward the sink.

Reserving a dedicated cell with each neighbor is required here to measure practically the unicast PDR. In a real deployment, the target node would reserve dedicated cells with its preferred parent only.

We use the default values for the different parameters, as depicted in the Table 1. Each experiment lasts 24 hours, logging approximately 132,000 broadcast and 140,000 unicast packets [¶].

4.3 Correlation Factor Analysis

First we quantify the correlation for each neighbor (n) between:

1. **unicast PDR**: the ratio of unicast packets transmitted to n and for which the transmitter receives an acknowledgement;
2. the **broadcast rate** (BR): the number of broadcast packets received from n during a certain time window.

[§] https://github.com/rodrigoth/openwsn-fw

[¶] Our dataset is available at https://github.com/rodrigoth/ewsn2018 and can be freely exploited by the community.

Table 1. Default values used in the correlation and ranking analysis.

Experiment	Duration	24 hours
	Topology	Multipath
	# of nodes	11
	Testbed side	Grenoble
TSCH	Slotframe length	31
	# of shared cells	1
	Timeslot duration	15 ms
	EB period	15 s
	Schedule policy	fixed
CoAP	CBR	1 pkt/sec
RPL	DAO period	60 s
	DIO period	10 s
Radio	802.15.4 channels	11-26
	Transmission power	-4 dBm

We measure here the Pearson correlation metric. All the symbols used in this work are depicted on Table 2.

The Pearson coefficient (Equation 3) is a linear correlation factor commonly used in statistics to measure the linear correlation between two inputs variables. It is defined formally by:

$$r_{x,y} = \frac{cov(X,Y)}{\sigma(X)\sigma(Y)} \quad (3)$$

with $\sigma(X)$ denoting the standard deviation for the stochastic variable X, and $cov(X,Y)$ the covariance of the variables X and Y.

This coefficient ranges from -1 to 1, where -1 indicates a perfect negative correlation and 1 indicates a perfect positive correlation. The correlation of 0 means the absence of any correlation between X and Y.

We aim to investigate how much time is required to identify a correlation. We adjust consequently from 1 to 10 minutes the observation window w during which we measure the unicast PDR and the broadcast rate. Because the PDR and the broadcast rate are stochastic variables, we need a sufficient observation window to have an accurate estimation. A small w means that the network can quickly react to changes, by identifying a significant change in the value of the stochastic variables.

We focus now on the approach used to calculate the correlation coefficient between the number of broadcast packets received and the dedicated cell reception rate. We divide the dataset in portions of w minutes. During a given portion, we compute the number of broadcast packets sent by a neighbor n (eq. 4) and received by the target node, likewise the PDR from the target node to this neighbor (eq. 5).

$$broadcast\ packets(n) = \sum [EB(n) + DIO(n)] \quad (4)$$

$$PDR(a,n) = \frac{\sum [ack(n \rightarrow a)]}{\sum [packets(a \rightarrow n)]} \quad (5)$$

Table 2. All the symbols used in this work.

Symbol	Description
n	n^{th} neighbor node
w	length of the observation window (time during which the different metrics are measured and averaged)
$EB(n)$	Nb. of Enhanced Beacons (EBs) received from the neighbor n
$DIO(n)$	nb. of DIOs received from the neighbor n
$ack(b \rightarrow a)$	Number of acks received from node b by node a
$packets(a \rightarrow b)$	Number of unicast packets sent to node b from node a
r	Pearson correlation coefficient
ρ	Spearman ranking correlation coefficient
f	Fisher coefficient
R_u	Unicast rank (i.e. ranking obtained when the neighbors are ordered by their unicast PDR)
R_b	Broadcast rank (i.e. ranking obtained when the neighbors are ordered by their broadcast reception rate)
$PDR(R_u, v)/PDR(R_b, v)$	PDR of the node in the v^{th} position in the unicast/broadcast ranking

With the results obtained from Equation 4 and 5, we calculate the Pearson coefficient r for every portion of time contained in the dataset (eq. 3). We obtain finally a set composed of all Pearson coefficients:

$$C = \{r_a, r_b, ..., r_{z-1}, r_z\} \tag{6}$$

Before calculating the average correlation, we apply the Fisher transformation to all the elements in C (eq. 6), which gives us the set F (eq. 7). This transformation reduces the bias of the average when working with repeated measurements [27].

$$F = \{f_a, f_b, ..., f_{z-1}, f_z\} \tag{7}$$

where f_i is calculated using the Equation 8.

$$f_i = 0.5 * ln\left(\frac{1+r_i}{1-r_i}\right) \tag{8}$$

To calculate the average of all correlation coefficients, we apply Equations (9) and (10), which consist in taking the average of all normalized coefficients and then converting them back to r scale.

$$\bar{F} = \left(\frac{f_a + f_b, ..., +f_{z-1} + f_z}{length\ of\ F}\right) \tag{9}$$

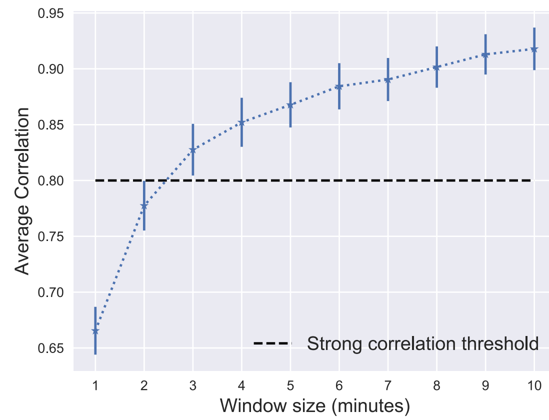

Figure 6. Average correlation for different observation windows, considering a confidence level of 95%.

$$\bar{r} = \frac{e^{2\bar{F}} - 1}{e^{2\bar{F}} + 1} \tag{10}$$

Figure 6 shows the average correlation for different window sizes (w). We observe that when the size of w is small (less than 2 minutes), the correlation between the two variables is weak. Indeed, the collisions represent a stochastic variable. Since broadcast transmissions are not so frequent (DIOs are sent every 10 seconds), the broadcast PDR is mis-estimated, highlighting a loose correlation. This phenomenon is even exacerbated by the slow channel hopping approach: the PDR may depend heavily on the physical channel. With a small number of packets, the broadcast transmissions are in this case not uniformly distributed among all the channels, creating a statistical bias.

To calculate the ideal size of w, we define the value $\bar{r}$ above which the correlation is assumed to be strong enough. Evans [28] considers a threshold value of 0.8 is an accurate choice. Indeed, for $r = 0.8$, 64% of the variation between the two variables can be explained by their relationship. The other 36% are due to external factors or sampling error.

In the evaluated topology, we note a strong correlation in a multichannel network from 3 minutes onwards. This result indicates a high positive correlation between the PDR and the number of received broadcast packets.

4.4 Neighbors Ranking Correlation

We aim here to evaluate the ability of the broadcast rates to identify a correct *ranking* of neighbors. Intuitively, the neighbor with the highest broadcast rate should also provide the best unicast quality.

We use the Spearman coefficient to evaluate the quality of such ranking. This coefficient (Equation 11) is a nonparametric measure between two ranked variables: their values are sorted based on a specific criteria. We use the PDR of unicast packets for the first ranking, and the broadcast rate of DIOs and EBs for the second ranking. The Spearman coefficient has the same range as the Pearson coefficient and

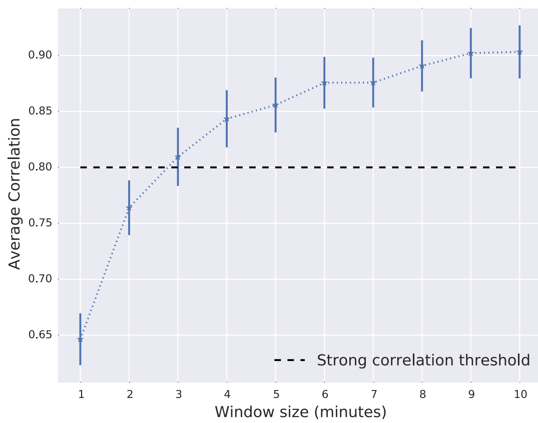

Figure 7. Spearman rank average correlation for different observation windows, considering a confidence level of 95%.

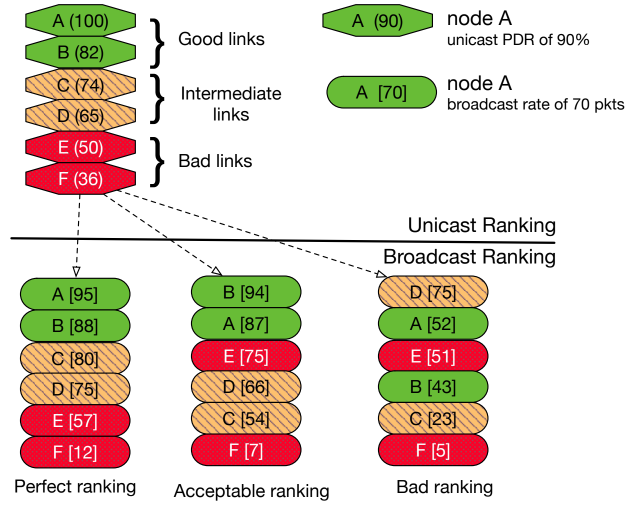

Figure 8. Different types of ranking agreement

exhibits the highest value when both variables have identical ranking and the lowest when they are fully opposed.

$$\rho_{x,y} = 1 - \frac{6\sum[rank(x_i) - rank(y_i)]^2}{l(l^2-1)} \tag{11}$$

where l is the sample size.

Figure 7 shows the average rank correlation for different values of w. We adjust the observation window from 1 to 10 minutes. We adopt the same approach (Fisher transformation) as in Section 4.3. We also consider that any value of $\bar{\rho}$ greater than 0.8 is a strong correlation factor (Section 4.3). As seen in the Pearson correlation result (Figure 6), for lower values of w the ranking correlation ρ is also weak. In this case, the collisions bias the estimation as well. Around 3 minutes, the two rankings exhibit a strong correlation.

4.5 Neighbors Ranking Efficiency

We consider that a node does not need to estimate initially the exact PDR of unicast packets to a neighbor. A node has rather to identify one good neighbor to join the network and to start reserving cells and sending packets. Thus, neighbors with almost the same link quality should be considered equivalent: a refined estimation would be too expensive for the corresponding added value. Inversely, a node must absolutely avoid selecting a bad parent: the joining procedure would be very expensive.

We make the following distinctions between different rankings (e.g. as depicted in Figure 8):

1. **Perfect ranking**: both rankings are identical. While the PDR are different in broadcast and unicast, all the neighbors are ranked identically;
2. **Acceptable ranking**: the rankings are slightly different, but the top ranked nodes are still neighbors with the best qualities. For instance, the node B is identified as a good neighbor when considering the unicast PDR;
3. **Bad ranking**: both ranking are very different, and we cannot use the broadcast PDR to identify good neighbors. In Figure 8, we would select the node D since it provides the highest broadcast rate although it provides a very poor unicast PDR.

We aim here to quantify the difference among the broadcast and unicast rankings. For instance, if we select the neighbor with the highest broadcast rate (our passive solution), what would be the decrease in PDR compared with a solution measuring directly the unicast PDR (which reserves a dedicated cell with each neighbor and generates probe packets)? We define the following metric to compute the normalized difference between the PDR which may be obtained using the two corresponding rankings (unicast vs. broadcast):

$$\Delta(R_u, R_b, t) = \frac{\sum_{v=1}^{t} PDR(R_u, v) - PDR(R_b, v)}{\sum_{v=1}^{t} PDR(R_u, v)} \tag{12}$$

We consider that a node must be able to identify two good preferred parents. Indeed, several routing solutions rely on a primary and a backup route to provide high-reliability. For instance, 6TiSCH advocates the selection of two parents, with a replication scheme among the two routes [29]. Thus, we aim here to evaluate the ability of the broadcast rate metric to select the two best ranked neighbors ($t = 2$ in eq. 12).

We can note that, in Figure 9, the $\Delta(R_u, R_b, 2)$ tends to decrease for higher values of w. Around $w = 3$ minutes, its average yields less than 0.1, corresponding to a very low error rate: the broadcast rate can be used successfully to infer the unicast quality.

Next, we measure the impact of the density, i.e. from 10 to 19 neighbors. When the density is high, the collision rate in the shared slots increases. Thus, the broadcast rate is more loosely dependent on the link quality: the packets are dropped mostly because of collisions.

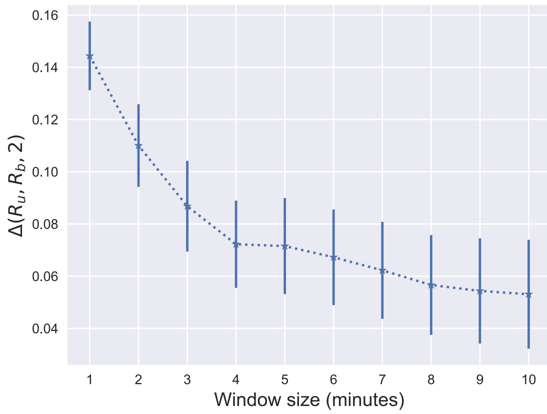

Figure 9. Average of the difference between the top ranked nodes in both rankings (unicast and broadcast) for different observation windows (w), considering a confidence level of 95%.

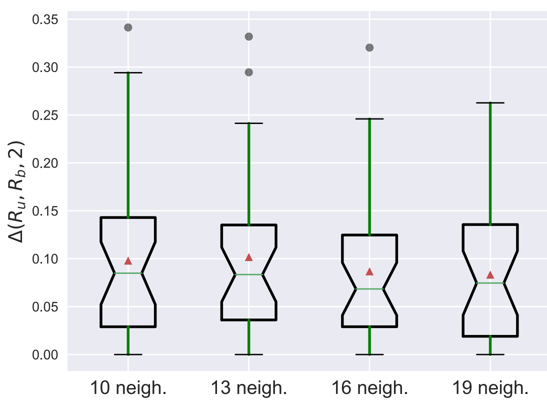

Figure 10. Impact of the number of neighbors on the ranking, considering a confidence level of 95%.

To limit the collision probability for this quite high density, we use three shared cells, placed uniformly in the slotframe. Otherwise, the number of neighbors is too high and one shared cell would not be sufficient to transmit all the control packets [30].

Figure 10 shows the impact of the node density on our ranking approach for $w = 4$ minutes. We observe that the ranking performs similarly regardless of the number of neighbors. For such scenarios, the quantity of shared cells and their new positioning in the slotframe minimizes the impact of collisions on the ranking and we achieve a very low error rate. To deal with even higher density, we would have to increase the number of shared cells so the nodes have more opportunities to send their broadcast packets at different times. A collision of the broadcast packet would impact the convergence and the stability of the network [24].

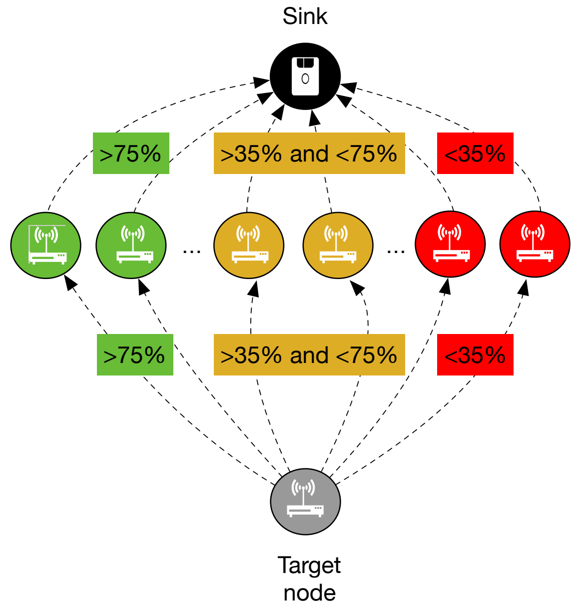

Figure 11. Multipath topology to measure the end-to-end path quality.

5 Using the Broadcast Rate for the RPL's rank

6TiSCH minimal advocates the use of the Objective Function OF0 [23] for rank computation using the ETX metric to estimate the link quality. Straightforwardly, we can use the rate of broadcast packets received instead of the ETX in the rank computation:

$$rank(i,j) = rank(j) + 3 * \left[\frac{expected\ adv.\ packets\ by\ i}{DIO(j) + EB(j)} \right] * 256 \quad (13)$$

The rank of a node is in this case the rank of the preferred parent, plus the inverse of the broadcast rate. To normalize the broadcast rate, a node assumes its broadcast period is the same as those of its neighbors.

To measure the accuracy of this rank to capture the end-to-end path quality, we focus now on a multipath topology (Figure 11). Each link has a different link quality, which is roughly the same for each hop of the path. A good path must provide a high end-to-end reliability, with a small delay. We argue that the broadcast rate allows each node to select the best parents, and to set up globally efficient routes.

The broadcast rate is simply obtained by monitoring the number of broadcast packets received from each neighbor along each path. Then, we measure independently the end-to-end PDR of each possible path: dedicated cells are reserved for each path, and the target node generates one data packet per second to the sink. We selected an observation window of 4 minutes since it exhibits a high linear correlation (Figure 6) and a high rank correlation (Figure 7) (the confidence interval is above 80%).

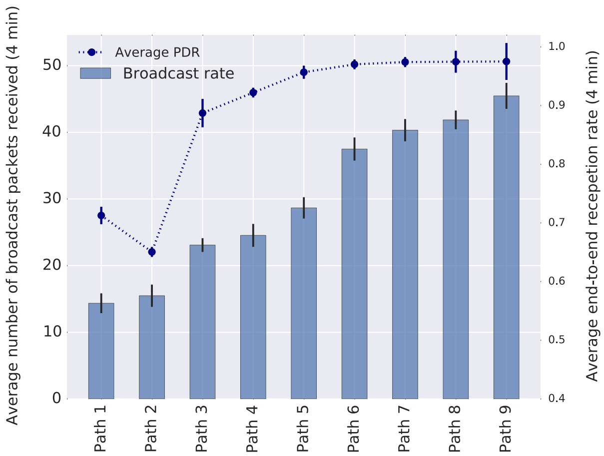

Figure 12. Unicast PDR and Broadcast rate for each path (confidence level of 95%).

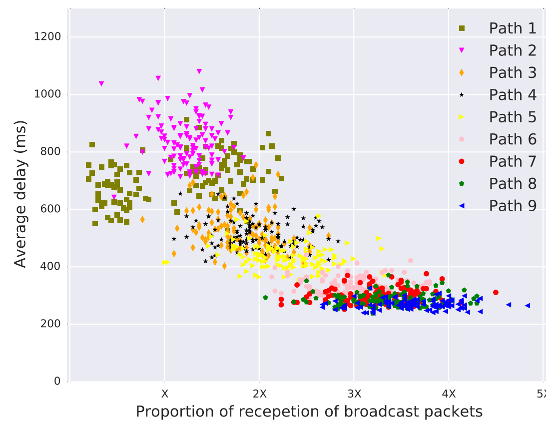

Figure 13. Impact of the number of broadcast rate on the end-to-end delay.

Figure 12 illustrates the broadcast rate of each neighbor and the PDR of each path, ordered by their path quality.

Both paths 1 and 2 exhibit the lowest broadcast rate and the poorest end-to-end PDR – ≈ 65%. We have then the medium quality paths (3 and 4) which provide a medium end-to-end PDR and they are classified medium for the broadcast rate metric. Finally, all the other paths provide an high PDR, with significantly different broadcast rates (37 for the path 9 vs. 23 for the path 5). While we are not able to derive the PDR from the broadcast rate, it remains a very good discriminator of path quality. Practically, a node has several good possible parents and should select one of them, while limiting the overhead (probes, number of 6P reservations, parent changes, etc.). In conclusion, the broadcast rate is definitely an accurate metric to detect fast and passively one of the best parents.

However, the most important is that the highest broadcast rate allows the target node to identify the best path, which is actually the case.

We finally measure the end-to-end delay achieved through the different paths (Figure 13). We clearly identify a strong correlation between the PDR and the delay: a low reliability implies that the node has to retransmit several times the packet before receiving an acknowledgement. Thus, as anticipated, paths 1 and 2 present the highest delay and the poorest reliability.

6 Integration in 6TiSCH and SF0

We integrated our passive approach in the latest version of OpenWSN stack[‖]. Each nodes registers every reception of a broadcast packet sent by each of its neighbors for a period of time (equal to the observation window *w*). Every *w* minutes, the nodes ranks its neighbors based on the broadcast rate, and selects the best one as its preferred parent. In addition, we activate the Scheduling Function SF0 [11] to allocate the dedicated cells in a distributed manner.

[‖]http://openwsn.org/

We compare the two following approaches:

- **blind selection**: the default behavior of openWSN, using a default ETX metric of 2 (50% of error rate) for an inactive neighbor;
- broadcast-rate **aware selection**: based on the observed broadcast rate, a node selects as preferred parent the neighbor with a smaller rank and with the highest broadcast rate.

To study the convergence, we compare the number of parent changes for both versions. Every parent changing starts a new negotiation process between a node and its new preferred parent, where they exchange control packets to reserve bandwidth. The objective here is to show that our method is tolerant to short variation on the link quality.

We use a peer-to-peer network composed of 13 nodes deployed in the same corridor in the FIT/IoT-LAB in Grenoble. Unlike the others experiments, we did not force any topology here: the nodes decide autonomously the final topology and the dedicated cells to use. The slotframe length was 101 slots, with 5 shared cells. In our implementation, we positioned uniformly the shared slots in the slotframe and we restrict them to broadcast packets only, as we did in the previous experiments. Each experiment lasts for 1 hour and the observation window was 4 minutes.

Figure 14 shows the Complementary Cumulative Distribution Function (CCDF) of the number of parent changes for both implementations.

We can note that the blind selection has a higher rate of parent switching. Because a node does not effectively compute the link quality with an inactive neighbor, it uses the default value cost to estimate the link quality (see Section 3.1). This implicitly makes a node to “blindly” select another parent when the link between this node and its preferred parent is refined and classified as bad. On the other hand, our passive approach divides approximately by half the number of parent switchings.

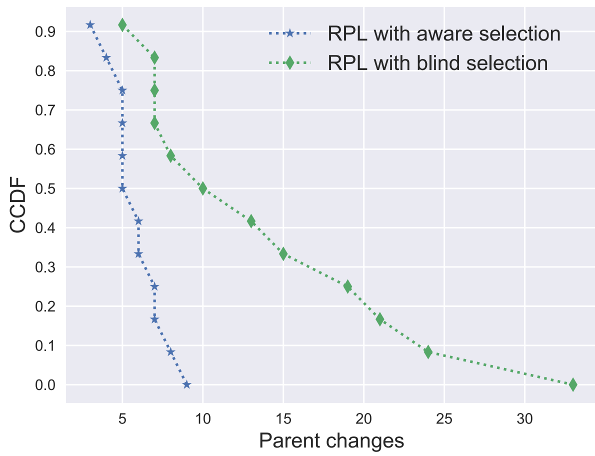

Figure 14. The difference in number of preferred parent changes between the two implementations.

7 Lessons Learned

Sending probe packets in a 6TiSCH network is unrealistic: it may consume a huge volume of resource to test each node individually and to select the most accurate ones. From the experimental results we collected, our approach has showed that estimating the link quality using the broadcast reception rate is a viable solution for 6TiSCH networks. With our passive method, a node can identify the *best* neighbors, i.e. those which would provide a high PDR and a low delay.

We also draw the following conclusions:

Observation window size: straightforwardly, the accuracy of our method is closely linked to the length of the observation window. Since packet losses represent a stochastic variable, a too short observation window implies a larger bias and the node would misestimate the link quality. This problem is even exacerbated by the *multichannel effect*: each channel must be used a sufficient and equal number of times to have a correct estimation since some channels may be subject to external interference [5]. However, a longer observation window reduces the reactivity time and a node may not react fast enough when a sudden fault happens. In our experiments, 4 minutes represented a good tradeoff.

Rapid convergence: our method minimizes the number of parent changes during both the bootstrap and the operating phases. A passive estimation is required to take fast decisions, without generating a large overhead. As a result, our approach has shown to be energy efficient and a node will change its preferred parent only when necessary.

Collisions minimization: because our method exploits the shared cells, we need to minimize the occurrence of collisions in these slots. One option is to restrict the shared cells to broadcast packets only and to reduce the slotframe length. With a shorter slotframe, the shared cell will repeat more often giving the nodes more chances to transmit their advertisement packets at different times. A second one is to uniformly distribute the shared cells along the slotframe, instead of placing them together at the beginning. By applying the second option, our method showed robustness to be used in a network with high density (up to 19 neighbors).

Scheduling integration: our method may be plugged in both centralized and distributed scheduling approaches. In a centralized approach, each node may compute a ranking of its neighbors, pushed to the Path Computation Element (PCE) which would be in charge of selecting the best routes and to construct the final schedule. This monitoring information may be continuously transmitted to the central component by a combination of piggybacking and dedicated control packets [31]. Different broadcast rates and rankings may be a good indicator for the PCE to detect asymmetrical links, and to allocate accordingly enough bandwidth along the best routes.

8 Conclusions and Perspectives

We proposed here to monitor the broadcast rate to estimate the unicast link quality of the different neighbors. Rather than estimating precisely the unicast PDR, we aim to *classify* the different neighbors, to only select as preferred parent one of the best neighbors. Indeed, the unicast PDR and the broadcast rate exhibit a very strong correlation, and we highlighted experimentally that we could safely rank the link qualities using this metric. We integrated this mechanism in the 6TiSCH stack, and we demonstrated experimentally the relevance of this passive method. We reduced the number of parent changes because a node selects immediately one good preferred parent, instead of testing iteratively each of its neighbors.

In future work, we expect to study the impact of a variable broadcast rate, particularly due to the trickle timer of RPL. Indeed, different nodes may use a different DIO period, because of the trickle algorithm (the period is doubled when no inconsistency is detected). However, we conjecture that the trickle timer should not impact the convergence during the bootstrapping period: all the nodes will reset their DIO period until the routing layer has converged. Inconsistencies may rather arise locally when the network is stable, but different neighbors use different DIO periods. We aim to quantify such impact.

9 References

[1] Martin Wollschlaeger, Thilo Sauter, and Juergen Jasperneite. The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0. *IEEE Industrial Electronics Magazine*, 11(1):17–27, 2017.

[2] Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011). Low-Rate Wireless Personal Area Networks (LR-WPANs). Technical report, IEEE, April 2016.

[3] F. Righetti, C. Vallati, G. Anastasi, and S. Das. Performance Evaluation the 6top Protocol and Analysis of its Interplay with Routing. In *Proceedings of IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 1–6, May 2017.

[4] Emilio Ancillotti, Carlo Vallati, Raffaele Bruno, and Enzo Mingozzi. A reinforcement learning-based link quality estimation strategy for RPL and its impact on topology management. *Elsevier Computer Communications*, 112:1 – 13, 2017.

[5] Thomas Watteyne, Ankur Mehta, and Kris Pister. Reliability through frequency diversity: Why channel hopping makes sense. In *Proceedings of ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)*, 2009.

[6] ISA-100.11a-2011:. Wireless Systems for Industrial Automation:Process Control and Related Applications. *International Society of Automation (ISA) Std.*, 1, May 2011.

[7] WirelessHART. Specification 75: TDMA Data-Link Layer. HART Communication Foundation Standard, 2008.

[8] T. Watteyne, M. Palattella, and L. Grieco. Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement. RFC 7554, 2015.

[9] Rodrigo Teles Hermeto, Antoine Gallais, and Fabrice 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.

[10] Q. Wang, X. Vilajosana, and T. Watteyne. 6top protocol (6p). draft, IETF, October 2017. draft-ietf-6tisch-6top-protocol-09.

[11] D. Dujovne, LA. Grieco, MR. Palattella, , and N. Accettura. 6TiSCH 6top Scheduling Function Zero (SF0). Internet-draft, IETF, July 2017. draft-ietf-6tisch-6top-sf0-05.

[12] X. Vilajosana and K. Pister and T. Watteyne. Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e (6TiSCH) Configuration. RFC 8180, IETF, February 2017.

[13] Nouha Baccour, Anis Koubâa, Habib Youssef, and Mário Alves. Reliable link quality estimation in low-power wireless networks and its impact on tree-routing. *Elsevier Ad Hoc Networks*, 27:1–25, 2015.

[14] S. Dawans, S. Duquennoy, and O. Bonaventure. On link estimation in dense RPL deployments. In *Proceedings of IEEE Conference on Local Computer Networks*, pages 952–955, Oct 2012.

[15] J. Ko and M. Chang. MoMoRo: Providing Mobility Support for Low-Power Wireless Applications. *IEEE Systems Journal*, 9(2):585–594, June 2015.

[16] A. Bildea, O. Alphand, F. Rousseau, and A. Duda. Link quality estimation with the Gilbert-Elliot model for wireless sensor networks. In *Proceedings of IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 2049–2054. IEEE, Aug 2015.

[17] Carlo Vallati, Emilio Ancillotti, Raffaele Bruno, Enzo Mingozzi, and Giuseppe Anastasi. Interplay of Link Quality Estimation and RPL Performance: An Experimental Study. In *PE-WASUN*, pages 83–90. ACM, 2016.

[18] L. Pradittasnee, S. Camtepe, and Y. C. Tian. Efficient Route Update and Maintenance for Reliable Routing in Large-Scale Sensor Networks. *IEEE Transactions on Industrial Informatics*, 13(1):144–156, Feb 2017.

[19] Ruan D. Gomes, Diego V. Queiroz, Abel C. Lima Filho, Iguatemi E. Fonseca, and Marcelo S. Alencar. Real-time link quality estimation for industrial wireless sensor networks using dedicated nodes. *Elsevier Ad Hoc Networks*, 59:116 – 133, 2017.

[20] Bogdan Pavkovic, Fabrice Theoleyre, Dominique Barthel, and Andrzej Duda. Experimental Analysis and Characterization of a Wireless Sensor Network Environment. In *Proceedings of ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)*, pages 25–32. ACM, 2010.

[21] D. Liu, Z. Cao, Y. Zhang, and M. Hou. Achieving Accurate and Real-Time Link Estimation for Low Power Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*, 25(4):2096–2109, Aug 2017.

[22] Rodrigo Fonseca, Omprakash Gnawali, Kyle Jamieson, and Philip Levis. Four-bit wireless link estimation. In *HotNets-VI*. ACM, October 2007.

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

[24] 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.

[25] V. Kotsiou, G. Papadopoulos, P. Chatzimisios, and F. Theoleyre. Is Local Blacklisting Relevant in Slow Channel Hopping Low-Power Wireless Networks? In *Proceedings of International Conference on Communications (ICC)*, pages 1–6. IEE, May 2017.

[26] Hyung-Sin Kim, Hosoo Cho, Hongchan Kim, and Saewoong Bahk. DT-RPL: Diverse Bidirectional Traffic Delivery through RPL Routing Protocol in Low Power and Lossy Networks. *Elsevier Computer Networks*, 126:150–161, October 2017.

[27] David M. Corey, William P. Dunlap, and Michael J. Burke. Averaging Correlations: Expected Values and Bias in Combined Pearson rs and Fisher's z Transformations. *The Journal of General Psychology*, 125(3):245–261, 1998.

[28] J.D. Evans. *Straightforward Statistics for the Behavioral Sciences*. Brooks/Cole Publishing Company, 1996.

[29] Pascal Thubert. An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4. Internet-draft, IETF, August 2017. draft-ietf-6tisch-architecture-12 (Work in Progress).

[30] Fabrice Theoleyre and Georgios Z. Papadopoulos. Experimental Validation of a Distributed Self-Configured 6TiSCH with Traffic Isolation in Low Power Lossy Networks. In *ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM)*, pages 102–110, 2016.

[31] G. Gaillard, D. Barthel, F. Theoleyre, and F. Valois. Monitoring KPIs in synchronized FTDMA multi-hop wireless networks. In *Proceedings of Wireless Days (WD)*. IEEE, March 2016.