



# Is Link-Layer Anycast Scheduling Relevant for IEEE802.15.4-TSCH Networks?

Rodrigo Teles Hermeto, Antoine Gallais, Fabrice Theoleyre

## ► To cite this version:

Rodrigo Teles Hermeto, Antoine Gallais, Fabrice Theoleyre. Is Link-Layer Anycast Scheduling Relevant for IEEE802.15.4-TSCH Networks?. 44th IEEE Local Computer Networks Conference (LCN), Oct 2019, Osnabrück, Germany. 10.1109/LCNSymposium47956.2019.9000679 . hal-02336182

**HAL Id: hal-02336182**

**<https://hal.science/hal-02336182>**

Submitted on 28 Oct 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Is Link-Layer Anycast Scheduling Relevant for IEEE802.15.4-TSCH Networks?

Rodrigo Teles Hermeto<sup>‡</sup>, Antoine Gallais<sup>‡§</sup> and Fabrice Théoleyre<sup>‡</sup>

<sup>‡</sup>ICube Laboratory, CNRS, University of Strasbourg, France

email: lastname@unistra.fr

<sup>§</sup>FUN - Self-organizing Future Ubiquitous Network, Inria Lille - Nord Europe, France

**Abstract**—With the wide adoption of low-power wireless transmissions, industrial networks have started to incorporate wireless devices in their communication infrastructure. Specifically, IEEE802.15.4-TSCH enables slow channel hopping to increase the robustness, and relies on a strict schedule of the transmissions to increase the energy efficiency. Anycast is a link-layer technique to improve the reliability when using lossy links. Several receivers are associated to a single transmission. That way, a transmission is considered erroneous when none of the receivers was able to decode and acknowledge it. Appropriately exploited by the routing layer, we can also increase the fault-tolerance. However, most of the anycast schemes have been evaluated by simulations, for a sake of simplicity. Besides, most evaluation models assume that packet drops are independent events, which may not be the case for packet drops due to e.g. external interference. Here, we use a large dataset obtained through an indoor testbed to assess the gain of using anycast in real conditions. We also propose a strategy to select the set of forwarding nodes: they must increase the reliability by providing the most independent packet losses. We demonstrate using our experimental dataset that anycast improves really the performance, but only when respecting a set of rules to select the next hops in the routing layer.

**Index Terms**—IEEE 802.15.4-TSCH; anycast scheduling; experimental evaluation; correlation; packet losses independency

## I. INTRODUCTION

Industrial networks target mainly the monitoring of safety-related processes with strict guarantees. To reduce the deployment costs, and to make the system more flexible, industrial networks rely more and more on low-power wireless communication. Unfortunately, these technologies are known to be lossy with no delivery guarantee [1].

To cope with this limitation, standards have been proposed to provide determinism on top of an unreliable link layer, such as IEEE 802.15.4-TSCH [2]. This standard carefully schedules the transmissions such that each device has enough transmission opportunities to send a packet in a reliable way. In addition, slow-channel hopping allows the nodes to combat the effect of external interference.

IEEE802.15.4-TSCH supports both centralized and distributed scheduling algorithms [3]. In a centralized approach, a controller knows a priori the radio topology, and the volume of traffic generated by each device. On the other side, the network has to reactively allocate the transmission opportunities (i.e. the cells) to each transmitter when applying a distributed approach.

Anycast has been proposed to authorize several receivers to be active at the same time. Since a transmission is lost only if all receivers fail to receive a packet, the network reliability and the energy efficiency can be significantly improved [4]. Hosni *et al.* [5] investigated the impact on the reliability when choosing the best parents to forward the packets. However, they assume that packet loss probabilities are independent for all the links, which may not hold practically. For instance, external interference may impact all the receivers simultaneously [6].

In this paper, we conduct a thorough experimental study to assess the relevance of an anycast technique at the link layer to improve the network reliability. We use an indoor large-scale platform, mimicking a smart building application, where multipath increases the complexity. Besides, other colocated networks (Wi-Fi, or IEEE802.15.4 compliant networks) may also generate external interference.

The contributions of this paper are as follows:

- 1) we evaluate the correlation factor among the packet transmissions for a group of receivers. In other words, for each device, does there exist a set of neighbors with (almost) perfectly independent packet losses?
- 2) we explain a method to schedule several receivers for a packet by exploiting minimal 6TiSCH to schedule, to increase the packet delivery ratio for each transmission;
- 3) we show that greedily selecting the best parents (i.e. providing the higher Packet Delivery Ratio) is insufficient since they may exhibit very correlated statistics. We thus propose an heuristic to select a proper set of parents;
- 4) we assess the performance of our heuristic, evaluating the reliability achieved in a multihop, realistic environment.

## II. BACKGROUND & RELATED WORK

We detail how IEEE802.15.4-TSCH and 6TiSCH work together since we rely on these two standards for the performance evaluation. We also detail the related contributions in anycast scheduling, and on the independency characterization of packet losses.

### A. IEEE 802.15.4-TSCH & 6TiSCH overview

IEEE 802.15.4-2015 has proposed the TSCH mode, largely inspired from the previous ISA100.11a [7] and WirelessHART [8] standards. It relies on slow channel hopping to external

interference: one packet and its retransmissions use different radio channels (frequencies), and make the packet drops less repetitive.

TSCH defines a slotframe, which consists of a matrix of cells (i.e. timeslots and channel offsets). The standard defines two different types of cells :

- dedicated:** cells should be assigned to a group of non-interfering radio links. In that case, the transmitter does not implement any contention resolution algorithm since it considers it has a *full* access. It just triggers a CCA before the transmission to deal with external interference;
- 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 (ack) is required but not received, the transmitter considers a collision occurred. In that case, it selects a random backoff value, and *skips* the corresponding number of shared cells.

Many distributed and centralized scheduling algorithms have been proposed so far for slow channel hopping MAC [3]. Centralized approaches rely on a controller which computes a schedule, pushed to each device. Inversely, a distributed scheduling relies on an algorithm implemented in each node, so that any pair of nodes can negotiate the cells to use.

In addition, 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 defines a protocol which is in charge of modifying the schedule (i.e. 6P [9]). Typically, a centralized scheduler uses 6P to modify the set of cells, through end-to-end commands. Inversely, distributed scheduling allows a pair of nodes to use 6P to negotiate directly the cells to use.

6TiSCH also defines a minimal configuration (i.e. 6TiSCH-Minimal) where all the devices wake up synchronously during *shared* cells [10]. Slotted-Aloha is used to solve the contention. These cells are typically used by default to transmit the broadcast control packets, such as Enhanced Beacons (EB) or DIO (DODAG Information Objects).

### B. Anycast transmissions

Wireless transmissions are broadcast by nature and any neighbor of the transmitter may overhear outgoing transmissions. Several works have proposed to use such feature, for both synchronous and asynchronous approaches. With preamble sampling approaches, the transmitter sends some short preambles: the first node waking-up acknowledges the preamble and starts receiving the packet [11]. This concept is quite old and has already been proposed when considering opportunistically the routing and MAC layers together [12], or to allow fast neighbor discovery by mobile nodes with anycast [13]. Unfortunately, this approach does not increase the reliability in synchronized protocols: only one receiver would be awake at a time.

We focus here rather on synchronous MAC approaches. For instance, ISA-100.11a-2011 implements the concept of duocast where two receivers are assigned to the same transmission. The receiver with the highest priority, acknowledges first the packet when it decodes correctly the packet. When the primary receiver fails, the secondary node has the opportunity to acknowledge the packet, and to place it in its forwarding queue. Thus, we have immediately a fallback solution to reduce the transmission delay, and the required number of retransmissions.

In a general manner, Huynh *et al.* [4] investigate the interest of anycast when considering ideal conditions (the channel state information is known by the receiver and transmitter). They demonstrate that an optimal scheduling policy exists, where the same reliability is achieved with a lower energy consumption and delay when using anycast at the link layer.

In synchronized networks such as IEEE 802.15.4-TSCH, the transmitter replaces the link-layer destination by an anycast link-layer address. A set of receivers is then subscribing to this anycast address, waking-up simultaneously. Hosni *et al.* [5] demonstrate that picking the best parents (higher PDR) represents an optimal strategy when neglecting the overhead for notifying the transmitter the packet was correctly decoded. However, they assume that packet loss probabilities are independent for all the links, which may not hold practically.

### C. Packet Losses Correlation

While ideal radio propagation models provide very stable characteristics, experimental evaluations prove reality is much more complex [14]. The link burstiness measures the time-variant packet losses for a given link [15]. A receiver may succeed to decode a sequence of packets, and then stop receiving for a long time. In other words, the packet losses are not independent, because of e.g. external interference. Identifying long-term stable links may help to avoid oscillations, but reduces also the routing diversity [16].

The anycast mechanism presented previously works ideally only when the packet losses are uncorrelated. Unfortunately, the literature has proved so far that packet losses exhibit a high spatial correlation [17]. In particular, the  $\kappa$  metric tries to estimate the correlation of packet losses among different links [15]. This metric is used to decide which *anycast path* to choose, and how to implement network coding to improve the reliability. We rather consider only non cross-layer features, and focus on the anycast feature at the link layer, with synchronized MAC protocols. We also investigate the impact of slow channel hopping on this independency.

## III. ANYCAST IN A 6TiSCH STACK

6TiSCH has been initially designed for unicast transmissions: in the schedule, a cell is assigned to a pair of receiver / transmitter. We here describe how to modify this stack to enable anycast transmissions.

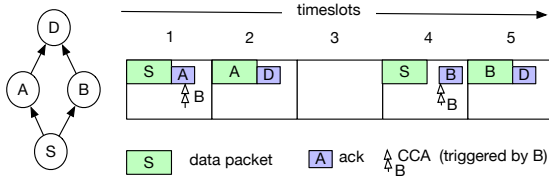


Fig. 1: Anycast Transmissions.

#### A. Limits of Unicast Communications

Because radio links are lossy, over-provisioning must be implemented, i.e. additional cells to retransmit the packets [18]. Unfortunately, these additional cells impact negatively:

**the delay:** the device has to wait for the next transmission opportunities. Since we implement a low duty cycle ratio, we may in the worst case wait for the next slotframe, which increases the delay and the jitter;

**the network capacity:** we expect a huge utilization of the same unlicensed band. Thus, we must use scarcely the radio resources: provisioning additional cells means also reducing the network capacity for the rest of the network, or the other co-located deployments. In particular, IEEE 802.15.4-TSCH has been proved to generate a large number of collisions in dense deployments [19].

Using a single path is also not fault-tolerant. A single faulty node is sufficient to *break* the route, and no packet is finally received by the controller. Some may argue that the routing protocol (aka RPL [20]) is in charge of finding another route after having detecting the fault. However, such reconfiguration is particularly expensive in synchronized networks: the bandwidth has to be re-allocated along the novel path. Since this reservation requires a large number of control packets, the convergence may be quite long, i.e. a few minutes in some cases, with possibly oscillations [21]. We are convinced that multiple *anycast paths* would make the routes much more robust and fault tolerant, while avoiding heavy reconfiguration costs (i.e. bandwidth, energy, time).

#### B. Implementing anycast

Anycast transmissions would reduce the number of transmissions by assigning several receivers for one single transmitter. We aim to provide the same level of reliability with less retransmissions. Let us consider the scenario illustrated in Figure 1. The node S has two parents A and B. When it transmits a data packet, A has the highest priority and sends its ack first. The second parent B triggers a CCA after a fixed interval, and detects in the first timeslot an ack is on-going: it drops the packet from S. Then, A forwards the packet to D. For the second packet of S (timeslot 4), the node A is unable to decode the packet, and does not detect anything when triggering its CCA: it sends its ack to S and finally releases the packet to D.

We can note that anycast is only efficient for the nodes more than 2 hops away from the border router. However, longer routes tend to be less reliable, because more relay nodes

have to forward the packets. Thus, anycast may improve the reliability and would in the worst case fallback to the unicast scenario.

Typically, anycast implies the following consequences:

**Receivers ordering:** because several receivers can decode the packet, the receivers have to be *prioritized*. A receiver will forward a packet only if all the other receivers with a larger priority failed to decode the packet;

**ACK:** the transmitter must be sure its packet has been received by at least one receiver. For this purpose, anycast often advocates the usage of contention resolution for ack. More precisely, a receiver waits for a backoff inversely proportional to its priority: if the CCA triggered before transmitting its ack is positive, this means that a receiver with a larger priority is currently transmitting its ack; it stops the process;

**False negative:** with high external interference, the nodes may conclude erroneously with a CCA that a ack is already in transmission. They would stop the process, and the transmitter will not receive an ack for its transmission, even if one of the receivers was able to decode it.

However, the transmitter would in that case retransmit the packet. Thus, the reliability with anycast would be at least as good as the unicast case;

**Duplicates:** if the acks collide (false negative for the CCA), the transmitter will retransmit its packet, generating duplicates. Thus, the set of forwarding nodes has to be properly constructed to avoid hidden terminals. Typically, each node has to report the list of its neighbors in its *beacons*, so that a node can select parents with a sufficient link quality from one to the others.

#### C. Shared Cells Scheduling

Anycast requires to let several receivers to wake-up synchronously, spending more energy. We want to show experimentally that such mechanism is really efficient to improve the reliability. We aim to prove that the packet losses may be sufficiently independent to provide a significant gain in end-to-end Packet Delivery Ratio (PDR). Anycast is energetically relevant when the gain is higher than the energy spent for overhearing.

Anycast scheduling imposes that a transmitter negotiates a common cell with all the receivers. Thus, several handshakes are required to pre-reserve a cell in each receiver (first step) and then start using it when all the receivers confirm it was available. Practically, this increases both the number of 6P packets and the convergence time. Our objective is here to rather quantify the gain of anycast, before modifying the protocols to support anycast. Thus, we propose here to use the 6TiSCH-minimal (Section II-A) schedule. A collection of shared cells are reserved for all the nodes. Thus, a transmitter can safely use shared cells: all its neighbors will be awake, and will be able to decode the packet if they have to forward it.

We focus here on quantifying the gain of using anycast scheduling. By generating a sufficiently low amount of data

packets, we can reduce the probability that the same shared cell is used by different transmissions (i.e. collisions). Thus, we can focus uniquely on the anycast mechanism. In a future work, we will investigate the reservation of multi-receivers dedicated cells to make the 6TiSCH stack fully anycast compliant.

#### D. Preliminary Results

To assess the degree of correlated losses, we collect a large dataset of packet transmissions / receptions on a large-scale testbed. Our objective here is to characterize the wireless links to verify the feasibility of using anycast communication in an indoor environment. We use the FIT IoT-LAB testbed in Lille. In this testbed, all nodes are disposed across 3 floors at the Inria building. Additionally, we can expect external interference originated from other wireless devices, such as Wi-Fi devices and/or concurrent 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. We configure a static schedule where each node takes a turn to broadcast a burst of 20 packets, every 30 seconds, during 15 minutes. We select randomly 15 nodes on the second floor of the testbed. Each node records the success / failure of each packet of the burst.

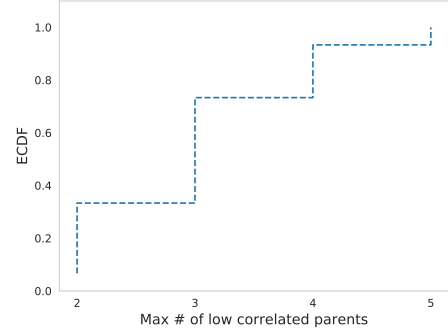
We compute the Pearson ( $\phi$ ) [22] correlation for every pair of links that received at least one packet from a transmitter in a given burst. This correlation factor is particularly relevant to measure the correlation among two stochastic variables which do not present the same average value. Indeed, we aim to compare the correlation between two links, whatever their average PDR is.

We consider all links with correlation factor below 0.4 as low correlated [23]. Figure 2a depicts the Empirical Cumulative Distribution Function (ECDF) of the number of low correlated parents by transmitter. We observe that all nodes had at least two *independent* parents, i.e. their packet losses are very loosely correlated. Thus, selecting these two parents would increase the Packet Delivery Ratio for the link: when the first parent fails to decode the packet, the secondary parent will independently try to decode it.

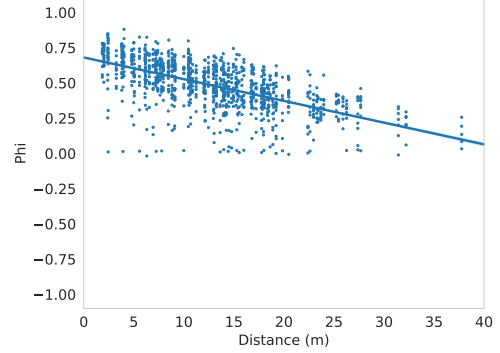
We also measure the impact of the distance on the inter-link correlation. Figure 2b presents a linear regression using the least squares method between the two variables considering 95 % of confidence level. As expected, longer links tend to present weaker correlation: when two receivers are farther than 25 meters, the average correlation is below our threshold value of 0.40. Thus, nodes geographically close to each other tend to present strong correlation and may not provide a significant gain in terms of diversity.

## IV. PARENT SELECTION FOR ANYCAST ROUTING

Inserting too many parents would waste energy: some of them would have to stay awake while never forwarding any packet (the other parents are sufficient). Inversely, we increase



(a) Low correlated parents.



(b) Linear regression between distance and Phi ( $\phi$ ) coefficient.

Fig. 2: (a) Maximum number of parents with low correlation by transmitter and (b) linear regression between distance and Phi with 95% of confidence level.

the number of retransmissions when too few parents are selected. Thus, we have to construct the right set of parents.

We consider here a convergecast traffic pattern. Each device selects multiple parents, and forwards all its packets in anycast to these parents.

#### A. Reliable Parent Set Notification

All the nodes have to wake-up during the shared cells. However, only the authorized receivers have the right to acknowledge and to forward a packet. In particular, the receivers have to know their *priority* for a given transmitter.

Since we consider industrial networks, the set of parents has to be stable to provide long-term performances. Each device has to notify its parent it has been selected, with its priority. This notification has to be reliable, to avoid deafness.

The ordered list of receivers (a short address with 16 bits) is transmitted in an Information Element, and piggybacked in the Enhanced Beacons and routing control packets. To the list is also associated a timestamp (Absolute Sequence Number), at which the transmitter will switch to the novel list.

The timestamp has to be sufficiently distant in the future to handle packet losses. For instance, with a 60% Packet Error Rate, the transmitter has to transmit on average 3

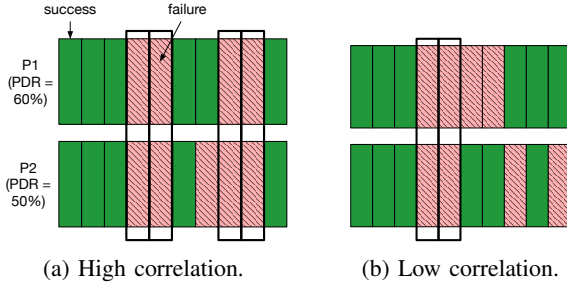


Fig. 3: Two links with different correlation values.

packets before switching to the novel list. Since we consider stable conditions [21], this list is seldom changed, limiting the overhead.

### B. Greedy PDR Parent Selection Limitation

Usually, we compute the Packet Delivery Ratio (PDR) to assess the reliability of a given transmission. In unicast communication, a node computes the PDR per link, i.e. the ratio of the number of ACKs and the number of packets transmitted to a specific neighbor.

If we assume independent packet losses, we can apply the optimal method described in [5]. Each node prioritizes its possible parents based on their RPL rank and their individual PDR. By using the RPL rank as condition, we ensure a loop-free route. We assume that nodes compute their respective ranks based on link quality metrics, such as Expected Transmission Count (ETX), to exploit reliable paths. Finally, a node selects greedily high-priority neighbors in its parent set until the Packet Delivery Ratio reaches a maximum value.

Let  $PDR(i, p)$  denote the Packet Delivery Ratio from the node  $i$  to its parent  $p$ , and  $\mathcal{P}(i)$  denotes the set of parents for  $i$ . A packet is considered lost if none of its parents received it. If we consider independent packet losses, we obtain:

$$PDR(i \rightarrow \mathcal{P}(i)) = 1 - \prod_{p \in \mathcal{P}} (1 - PDR(i, p)) \quad (1)$$

However, this equation does not hold anymore if packet losses are correlated among the different parents. To show the limitation of this approach, let us consider the Figure 3 illustrating the success / failure of 10 packets for two parents. In Figure 3a, we have two intermediate links with a PDR of 60 and 50% for parents 1 and 2 respectively. When we look at each individual transmissions, we observe that all packets received by Parent 2 were also received by Parent 1, exhibiting a strong correlation. The anycast delivery rate is in that case bounded by the highest PDR of the two links (60%). Formulated differently, anycast would not provide any gain, since the diversity is insufficient to decrease the packet losses.

Anycast scheduling can achieve higher reliability only if the receivers have weak or no correlation in terms of packet losses. This situation with the same PDR as in the previous example is highlighted in Figure 3b. In that case, anycast would achieve a

PDR of 80%, as only the 4<sup>th</sup> and 5<sup>th</sup> packets are not received by any of the two parents.

### C. Joint-Packet Delivery Ratio for anycast links

In realistic conditions, a packet may be lost because of e.g. external interference, which impacts all the receivers. In that case, we cannot anymore assume the packet losses are independent, and the equation 1 does not hold anymore.

Let us consider the Figure 3 illustrating the success / failure of 10 packets for two parents. Independent packet losses mean the conditional probability is nothing else but the product of the success probabilities. If the packet toward the parent P1 has been lost, the probability that P2 decodes the packet remains unchanged.

On the contrary, when packet losses exhibit a strong correlation, the conditional probability is significantly modified. Typically, if the packet toward P1 has been lost, this is highly probable that it has **also** been lost for P2. Inserting P2 in the forwarding set has no positive effect on the reliability: if P1 was unable to decode the packet, it is highly probable that P2 will never succeed to receive it either.

Since the parents are ordered by their PDR in the greedy-PDR strategy, the parents with a lower PDR have to be integrated in the parent set only if they provide independent results. Else, their reliability gain can be neglected.

Thus, we propose here the Joint Packet Delivery Ratio (**J-PDR**) metric to consider anycast transmissions with any packet losses correlation. J-PDR takes into account the fact that multiple receivers are listening for each transmission. Thus, we have to compute the multi-neighbor delivery ratio.

We use the concept of **transmission success sequences**. Each receiver stores independently the transmission success for the last  $k$  packets. This sequence of bits is reported regularly to the transmitter (within an Information Element as in [24]) so that it can compute the J-PDR.

More formally, let us denote by  $s_{ij}(k)$  the binary variable equal to 1 iff  $j$  has received the  $k^{th}$  packet from the node  $i$ . Thus  $\langle s_{ij} \rangle$  denotes the transmission success sequence from  $i$  to  $j$ . The node  $i$  computes J-PDR to its set of parents  $P$  as follows:

$$J\text{-PDR}(i \rightarrow P) = 1 - \frac{|\{k | \forall j, s_{ij}(k) = 0\}|_{k=1}^z}{|z|} \quad (2)$$

with  $z$  the number of packets in the sequence.

In other words, a packet is considered lost if none of the receivers received it properly.

### D. Greedy J-PDR Parent Selection

We propose here a greedy J-PDR Parent Selection that inserts  $k$  neighbors in the parents set while they increase the J-PDR for the link. More precisely, each receiver reports its reception sequence to the transmitter, i.e. a bitmap, with one bit per packet transmission. Since the transmitter uses a sequence number incremented at each packet transmission, a receiver knows a packet has not been received by identifying the *voids* in the sequence numbers.



The transmitter re-aligns all the sequences to update its parent set. The strategy considering greedily the parent according to their PDR is not anymore optimal. Thus, constructing the optimal set is computationally intensive: we have to test all the possible combinations.

We propose rather to implement a greedy approach using the J-PDR metric. We rank first the neighbors with a lower RPL rank according to a given metric (e.g. ETX, etc.). We select first the neighbor with the highest PDR, since it has a larger probability to increase the reliability for anycast. Next, we check for an additional neighbor that, when combined with the first selected neighbor, the J-PDR increases. Typically, neighbors with weak packet losses correlation have complementary transmissions: when one fails the other succeeds. We insert new neighbors in the parents set while the final J-PDR (selected parents and the candidate) for the link increases.

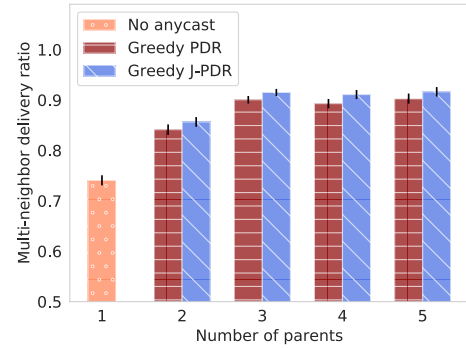
We aim here to *demonstrate experimentally* that anycast improves the reliability. Specific scheduling and routing strategies will be proposed in a future work.

## V. EXPERIMENTAL EVALUATION

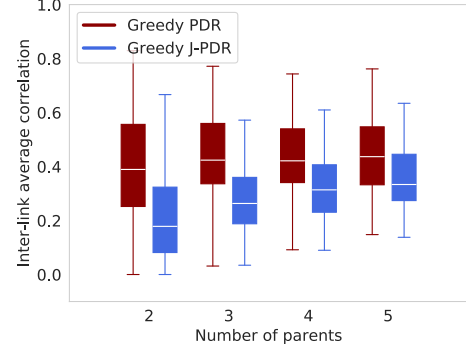
We base all our experiments on data traces obtained from a real deployment on the Lille testbed. In details, we deploy randomly 100 nodes across the building, and collect statistics for all transmitted packets and their respective sender/receivers. We use the same approach as in Section III-D, where a node takes a turn to broadcast a burst of packets periodically. The number of packets per burst and the broadcast periods are the same, 20 and 15 minutes respectively. The network operated for 25 hours during a regular working day, collecting approximately 7 million measurements. The complete dataset is freely available on Github\*.

Our objective is here to investigate the interest of using anycast at the link layer, apart from any specific protocol mechanism. Before spending some time to implement anycast on prototypes, we are convinced we need to prove anycast is efficient when exploiting real links, where packet losses may not be independent. Thus, we emulate a multihop topology using a custom IEEE 802.15.4-TSCH network simulator, exploiting directly the dataset. Indeed, we use directly the transmission successes / failures to determine if a packet is received by at least one receiver. By making our evaluation independent from any implementation, we aim to first prove this anycast mechanism needs to be investigated further. The code used in our experiments is also freely available on Github<sup>†</sup>.

We base our implementation on the 6TiSCH-minimal [10] to schedule multiple receivers for a given transmitter. To avoid collisions, we schedule different shared cells for each transmitter. Thus, each node has only one opportunity to transmit per slotframe repetition. We assume here that the schedule is pre-installed at the compilation time. We focus specifically on links with intermediate quality (e.g. < 75% of



(a) Multi-neighbor delivery ratio.



(b) Average correlation.

Fig. 4: Multi-delivery ratio and the average correlation for the two evaluated heuristics.

TABLE I: Experiments parameters.

<b>Experiment</b>	Traffic load	1000 packets
	# of nodes	100
	# Testbed	FIT-IoT Lab (Lille)
	Traffic pattern	Convergecast
<b>TSCH</b>	Slotframe length	101
	Timeslot duration	15 ms
	Transmissions attempts	4
	Schedule policy	6TiSCH minimal based
<b>Radio</b>	802.15.4 channels	11-26
	Transmission power	+3 dBm

PDR) to demonstrate how anycast transmissions can improve the network reliability.

### A. Multi-neighbor efficiency

We first consider a scenario where a node has to build its parents set among multiple candidate neighbors. In this scenario, each one of the 100 deployed nodes transmits 1,000 packets in a row, while the others register all receptions. We use the first 12 bursts (240 transmissions) for training: a node computes the individual PDR and the transmissions success sequences for each neighbor. These metrics are used to select the parents. Finally, the remaining 51 bursts (1,020 packets) are used to assess the long-term performance when using this static parent set. All the experiments parameters are depicted in the Table I.

\*<https://github.com/rodrigoth/anycast>

<sup>†</sup><https://github.com/rodrigoth/anycast/tree/master/simulator>

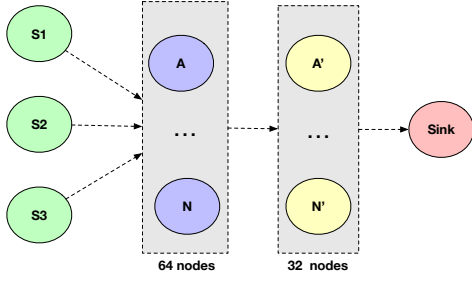


Fig. 5: Multi-hop topology composed of 100 nodes across the 3 floors of the testbed. The leaves generate 1000 packets destined to the sink.

We compare the performance of the two heuristics described in Section IV. Figure 4a reports the Packet Delivery Ratio obtained with the two heuristics (greedy PDR vs. j-PDR). Using a single parent (no anycast), allows the network to exploit links with a PDR of 75% on average. The two strategies perform exactly the same with one parent because they select as primary parent the neighbor with a lower rank, and with the highest PDR.

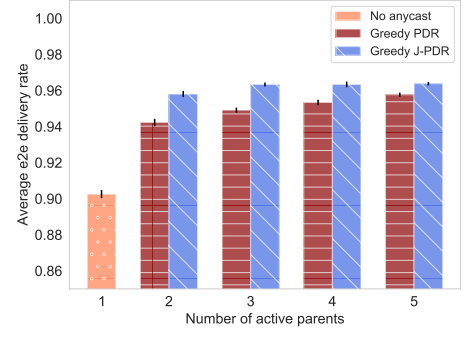
With 2 parents or more, anycast improves the PDR: the greedy PDR strategy achieves a PDR of 85%. Those results confirm that greedy J-PDR provides the highest reliability, since it selects the most independent set of parents. Greedy PDR tends to select neighbors that are geographically close to each other, since it considers only the PDR, and not the correlations. We can note that exploiting more than 3 parents has no benefit on the reliability, since only the firsts are effectively used.

To measure the ability of the strategies to select independent parents, Figure 4b reports the correlation factors of the different parents for a given node. In approximately 75% of the cases, all the selected parents have a weak correlation average among themselves (below 0.4). This low correlation highlights that our algorithm improves the spatial diversity and consequently reduces the number of retransmissions. On the contrary, the greedy PDR strategy may select non-independent parents in some cases, leading to a poor diversity (and thus, a lower reliability gain).

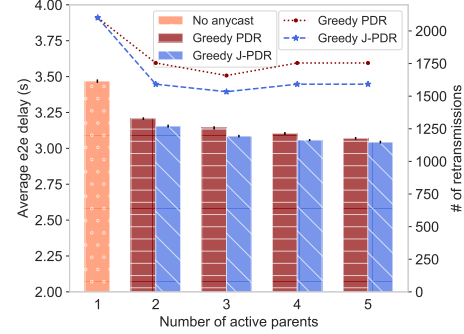
### B. End-to-end performance

We now consider a multi-hop network to focus on end-to-end performance. We place the sink at the extreme corner of the testbed to allow multi-hop transmissions. We rely here on RPL to construct the routes and to assign ranks to each node. We obtain typically the topology illustrated in Figure 5. To measure more precisely the gain of using anycast in multi-hop networks, we report only the end-to-end results for the devices which are 3 hops away from the sink (i.e. S1, S2 and S3).

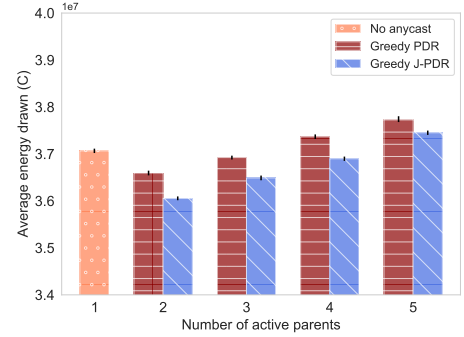
Figure 6a illustrates the end-to-end reliability achieved with anycast. Because anycast exploits multiple receivers, we increase significantly the end-to-end PDR when compared to the traditional unicast communication. Still, having more than



(a) E2E delivery rate.



(b) E2E delay.



(c) Energy.

Fig. 6: End-to-end performance on a multi-hop network for 30 simulations repetitions and 95% of confidence interval.

3 parents does only increase slightly the reliability. Greedy J-PDR keeps on providing the highest reliability, by selecting carefully independent parents, i.e. with independent packet losses. Here, anycast is efficient, and provides an end-to-end reliability of 96% when each device is authorized to retransmit at most 4 times the same packet.

Figure 6b reports the end-to-end delay. In particular, with only one parent, a packet is delivered after a large number of retransmissions, increasing mechanically the end-to-end delay. Indeed, with multiple parents, the nodes can exploit different links with variable conditions. Thus, a packet progresses on average *farther* with the same number of transmissions. With 2 parents, we reduce significantly the number of retransmissions and the delay. The difference between both heuristics becomes



more evident now: the higher spatial diversity added by the J-PDR heuristic reduces the number of retransmissions and consequently the end-to-end delay.

Finally, we measure the energy cost of using anycast, reported in Figure 6c. We use the energy model proposed in [25], which also targets IEEE 802.15.4-TSCH networks. This model computes the charge drawn per slot taking into account the amount of time that the CPU and radio are used. We assume that the radio is the main source of energy consumption and we consider only the energy spent in transmissions/receptions in our results. Additionally, we adjust the current values accordingly to our hardware (AT86RF231 radio). We use the timers from OpenWSN (<http://openwsn.org>) as reference to compute how long the radio is used in each timeslot.

Using anycast reduces the number of retransmissions and consequently the network saves energy. A retransmission impacts heavily the energy consumption, since in a shared cell all nodes wake-up simultaneously for a short time to listen for incoming packets, even if they are not scheduled. Thus a retransmission impacts not only the sender and the receiver(s), but all nodes in the network. Additionally, we can remark that the energy consumption is higher for the greedy PDR strategy: the parents are not independent, and some of them stay awake uselessly.

## VI. CONCLUSION & FUTURE WORK

We quantified here the interest of anycast to improve the reliability of IEEE 802.15.4-TSCH in a realistic indoor environment. In particular, we use experimental data to study in depth the correlation between packet losses. We proposed a heuristic to select a proper set of independent parents, able to improve the reliability. This greedy heuristic relies on the Joint Packet Delivery Ratio (J-PDR), denoting the actual PDR a link can provide, when considering non independent packet losses. This way, we reduce both the number of retransmissions, and the delay, when the source of the flow is several hops away from the destination. Our experimental results highlight that each device in our indoor environment has at least 2 parents with independent packet losses, i.e. with a  $\phi$ -factor inferior than 0.4.

After having proved here that anycast is efficient, we aim to propose in the future a novel anycast scheduling algorithm able to work with dedicated cells in 6TiSCH. Obviously, we can adapt any centralized scheduling algorithm, so that several receivers can be associated to a given cell (with the same transmitter). In distributed, we will propose to modify a Scheduling Function to take benefit from anycast. In particular, we will propose heuristics to obtain an accurate estimation of the packet losses, using a combination of passive measurements and probes, such as [26]. Finally, we expect also to implement the acknowledgements in the IEEE 802.15.4-TSCH stack, so that several (ordered) receivers can be associated with a single transmission.

## REFERENCES

[1] Tao Liu and Alberto E. Cerpa. TALENT: Temporal Adaptive Link Estimator with No Training. In *SenSys*, pages 253–266. ACM, 2012.

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

[3] 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.

[4] Thong Huynh, Fabrice Theoleyre, and Won-Joo Hwang. On the interest of opportunistic anycast scheduling for wireless low power lossy networks. *Computer Communications*, 104:55 – 66, 2017.

[5] I. Hosni and F. Theoleyre. Adaptive k-cast Scheduling for high-reliability and low-latency in IEEE802.15.4-TSCH. In *ADHOC-NOW*, volume 11104 of *LNCS*, pages 1–12. Springer, Sep 2018.

[6] Diego V. Queiroz, Marcelo S. Alencar, Ruan D. Gomes, Iguatemi E. Fonseca, and Cesar Benavente-Peces. Survey and systematic mapping of industrial wireless sensor networks. *Journal of Network and Computer Applications*, 97:96 – 125, 2017.

[7] ISA-100.11a-2011. Wireless systems for industrial automation: process control and related applications. International Society of Automation (ISA), 2011.

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

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

[10] 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.

[11] Simon Duquenooy, Olaf Landsiedel, and Thiemo Voigt. Let the tree Bloom: scalable opportunistic routing with ORPL. In *SenSys*, page 2. ACM, 2013.

[12] S. Kulkarni, A. Iyer, and C. Rosenberg. An address-light, integrated mac and routing protocol for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 14(4):793–806, Aug 2006.

[13] G. Z. Papadopoulos, V. Kotsiou, A. Gallais, P. Chatzimisios, and T. Noel. Low-power neighbor discovery for mobility-aware wireless sensor networks. *Ad Hoc Networks*, 48:66 – 79, 2016.

[14] Bogdan Pavkovic, Fabrice Theoleyre, Dominique Barthel, and Andrzej Duda. Experimental Analysis and Characterization of a Wireless Sensor Network Environment. In *PE-WASUN*, pages 25–32. ACM, 2010.

[15] K. Srinivasan, et al. The  $\kappa$  factor: Inferring protocol performance using inter-link reception correlation. In *MobiCom*, pages 317–328. ACM, 2010.

[16] Muhammad Hamad Alizai, Olaf Landsiedel, J6 Ágila Bitsch Link, Stefan Götz, and Klaus Wehrle. Bursty Traffic over Bursty Links. In *SenSys*, pages 71–84. ACM, 2009.

[17] 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.

[18] Felix Dobsław, Tingting Zhang, and Mikael Gidlund. End-to-End Reliability-aware Scheduling for Wireless Sensor Networks. *IEEE Transactions on Industrial Informatics*, 12(2):758–767, April 2016.

[19] S. Ben Yaala, F. Théoleyre, and R. Bouallegue. Cooperative resynchronization to improve the reliability of colocated IEEE 802.15.4-TSCH networks in dense deployments. *Ad Hoc Networks*, 64:112 – 126, 2017.

[20] T. Winter, et al. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550, IETF, March 2012.

[21] Rodrigo Teles Hermeto, Antoine Gallais, and Fabrice Theoleyre. On the (over)-Reactions and the Stability of a 6TiSCH Network in an Indoor Environment. In *MSWiM*, pages 83–90. ACM, 2018.

[22] Robert M. Gray and Lee D. Davisson. *An Introduction to Statistical Signal Processing*. Cambridge University Press, NYC, US, 2010.

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

[24] G. Gaillard, D. Barthel, F. Theoleyre, and F. Valois. Monitoring KPIs in synchronized FTDMA multi-hop wireless networks. In *Wireless Days*, pages 1–6, March 2016.

[25] G. Daneels, et al. Accurate energy consumption modeling of ieee 802.15.4e tsch using dual-band openmote hardware. *Sensors*, 18(2):437, 2018.

[26] Rodrigo Teles Hermeto, Antoine Gallais, Kristof Van Laerhoven, and Fabrice Theoleyre. Passive Link Quality Estimation for Accurate and Stable Parent Selection in Dense 6TiSCH Networks. In *EWSN*, pages 114–125, Madrid, Spain, 2018. ACM.