



Adaptive k-cast Scheduling for High-Reliability and Low-Latency in IEEE802.15.4-TSCH

Inès Hosni, Fabrice Theoleyre

► To cite this version:

Inès Hosni, Fabrice Theoleyre. Adaptive k-cast Scheduling for High-Reliability and Low-Latency in IEEE802.15.4-TSCH. International Conference on Ad Hoc Networks and Wireless (AdHoc-Now), Sep 2018, Saint-Malo, France. pp.3-14, 10.1007/978-3-030-00247-3_1 . hal-02336102

HAL Id: hal-02336102

<https://hal.archives-ouvertes.fr/hal-02336102>

Submitted on 6 May 2020

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.

Adaptive k-cast scheduling for high-reliability and low-latency in IEEE802.15.4-TSCH

Inès Hosni¹ and Fabrice Théoleyre²[0000–0002–7903–3520]

¹ LSC, University of Tunis El Manar, ines.hosni@hotmail.fr

² ICube lab, CNRS / University of Strasbourg, France, theoleyre@unistra.fr

Abstract. The Industrial Internet of Things tends now to emerge as a key paradigm to interconnect a collection of wireless devices. However, most industrial applications have strict requirements, especially concerning the reliability and the latency. IEEE802.15.4-TSCH represents currently a promising standard relying on a strict schedule of the transmissions to provide such guarantees. The standard ISA-100.11a-2011 has proposed the concept of duocast, where a pair of receivers are allocated to the same transmission opportunity to increase the reliability. In this paper, we generalize this approach to involve k different receivers, and we explore the impact of this technique on the performance of the network. We propose an algorithm assigning several receivers for each transmission to increase the probability that at least one device receives correctly the packet. By exploiting a multipath topology created by the routing layer, we are able to reduce the number of transmissions while still achieving the same reliability. We consequently increase the network capacity, and reduce significantly the jitter. Our simulation results highlight the relevance of this k-cast technique in TSCH for the Industrial Internet of Things.

Keywords: IEEE802.15.4-TSCH; k-cast transmissions; duocast; scheduling algorithms; high-reliability; low jitter

1 Introduction

Industry 4.0 is promised as the next industrial revolution, where digital automation aims to reduce the cost and to maximize the flexibility [11]. In particular, the Industrial Internet of Things (IIoT) aims to connect a large set of industrial objects to the Internet. In this context, a real-time infrastructure has to provide high-reliability for the wireless transmissions.

While the Internet of Things focused so far on *best-effort* solutions, industrial applications have often strict requirements concerning the reliability and the delay [2]. We need consequently specific MAC protocols able to provide strict guarantees. Deterministic approaches are particularly relevant, allocating a fixed bandwidth to each device or flow. These solutions can provide also flow isolation, where each flow receives a certain bandwidth, dedicated for *its* transmissions.

IEEE802.15.4-TSCH relies on channel hopping to increase the robustness to external interference and fading [1]. TSCH adopts a deterministic approach, and

schedules carefully the transmissions to avoid the collisions. More precisely, it allocates a set of different *cells* for interfering transmitters to reduce the contention while still avoiding collisions. A cell is defined by a pair of timeslot and channel offset. The channel offset is translated into a physical frequency at the beginning of the timeslot, to follow a pseudo-random sequence.

Many centralized and distributed scheduling algorithms [6] have been proposed so far for TSCH. To cope with unreliable links, the scheduling process often uses overprovisioning, and reserves several of cells so that a packet can be retransmitted. Unfortunately, more cells mean that the delay and the jitter may increase [13]. This over-provisioning also reduces the network capacity if the traffic is large and/or the links are very unreliable because of e.g. external interference.

IEEE802.15.4-TSCH is largely inspired from ISA-100.11a-2011 [9]. ISA has proposed a *duocast* mechanism to increase the reliability: two receivers are assigned to a given transmission. A transmission is considered a failure only if *both* receivers failed to decode correctly the packet.

In this paper, we aim to investigate the relevance of using k-cast scheduling, i.e. a cell is assigned to one transmitter, and k possible receivers (ordered by their priority). Subslots allow each receiver to acknowledge the transmission if none of the receivers with a larger priority sent an ack. The contributions of this paper are threefold:

1. we propose a scheduling algorithm tailored for k-cast transmissions. We propose to minimize the energy consumption for both transmitting and receiving the packets. Each node selects the set of parents to assign to a given cell to minimize this energy consumption;
2. we investigate the impact of the k-cast technique on the amount of duplicated packets. Indeed, the receivers may receive all the packets, but not the **ack** they transmit. In this case, the packets are duplicated wasting energy and bandwidth;
3. we implemented our scheduling algorithm and the k-cast technique to evaluate its performance. We highlighted this technique helps to improve the reliability with a limited impact on the energy consumption.

2 Related Work

IEEE 802.15.4-2015 has proposed the TSCH mode, which relies on a strict schedule of the transmissions [1]. The *slotframe* contains a fixed number of timeslots, during which at most one frame and its acknowledgment are transmitted. Each timeslot is labelled with an Absolute Sequence Number (ASN) which counts the number of timeslots since the PAN coordinator started. Based on the schedule, a node can decide its role (transmitter/receiver/sleeping mode) at the beginning of each timeslot.

IEEE 802.15.4-2015 TSCH implements a channel hopping approach to combat external interference and signal fading and, thus, to achieve high reliability [18]. For this purpose, each *cell* in the schedule is defined by a pair of timeslot

and channel offset. At the beginning of each timeslot, the actual frequency to use is derived from the channel offset and the ASN.

IEEE802.15.4-TSCH supports two medium access approaches:

shared cells are allocated to a group of nodes (e.g. for broadcast). A packet is for the first time transmitted without contention. If it corresponds to a unicast packet, and no **ack** is received, a random backoff is then used. This backoff corresponds to the number of cells to *skip* before the retransmission; **dedicated cells** are allocated to non-interfering transmitters. Thus, the transmitter just starts the transmission after a fixed offset from the beginning of the timeslot.

2.1 Improving the reliability

Several techniques have been proposed to improve the reliability in this kind of deterministic architecture. Over-provisioning consists in reserving additional cells to retransmit the packets if the transmission has failed. For instance, the number of cells may be inversely proportional to the Packet Delivery Ratio for the considered radio link [16]. Dobsław *et al.* [3] propose rather to fortify a centralized schedule by allocating additional cells to the most unreliable links until the deadline constraint cannot be anymore respected. Hashimoto *et al.* [5] propose to allocate shared cells for the retransmissions. However, shared cells are prone to collisions, impacting negatively the reliability of retransmissions.

Multipath also helps to improve the end-to-end reliability [10]. Papadopoulos *et al.* [13] propose to use multiple parents, replicating the packets over the two paths. With overhearing, several receivers try to decode a packet, increasing the reliability. A node which receives several times the same packet (identified with a sequence number) must drop the subsequent copies. However, the authors focus on a fault-tolerance scenario: the link quality drops suddenly on the primary path. We propose rather to address the *normal* case, where radio links may be unreliable *on average*.

Opportunistic routing consists in choosing several next hops. The destination address is transformed in anycast so that the same transmission can be received by any receiver. Typically, opportunistic routing helps to improve the efficiency in asynchronous wireless networks [17]. In IEEE802.15.4-TSCH, all the receivers wake-up synchronously. However, opportunistic routing is still efficient to improve the reliability: it is sufficient that *one* of the receivers decodes correctly the packet. Huynh *et al.* demonstrated that opportunistic scheduling helps to improve theoretically the reliability when exploiting a Rayleigh fading channel [7].

2.2 6TiSCH

6TiSCH has defined a set of protocols to execute IPv6 above IEEE802.15.4-TSCH. It relies on RPL (Routing over Low-Power Lossy Networks) to construct the routing topology [15]. A node uses a link quality metric and the rank of

its parent to compute its own rank, denoting its virtual distance from the sink. Typically, the link quality is estimated via ETX (Expected Transmission Count), the average number of transmissions before the packet is acknowledged by the receiver. By default, RPL uses only one preferred parent (next hop toward the sink), but has been extended to support also multi-parent routing [8].

Scheduling the transmissions has attracted much attention in the past [6]. The 6P protocol [14] is in charge of negotiating the cells to use. The transmitter sends a request to the receiver to decide how many, and which cells to use. Then, 6TiSCH relies on a so-called Scheduling Function (SF) to decide which and how many cells to use between each pair of nodes. 6TiSCH supports both centralized (e.g. TASA [12]) and distributed (e.g. SFx [4]) scheduling algorithms. For instance, SFx [4] maintains the number of cells at least equal to the number of packets to forward, and relies on an hysteresis function to limit the number of oscillations.

3 k-cast scheduling: choosing the right set of parents

Let us consider the example illustrated in figure 1. A node A has two ordered parents P_1 and P_2 . During the first timeslot, P_1 receives correctly the packet and acknowledges it. During the second one, the transmission to P_1 fails, but P_2 is able to decode the packet: it should be able to acknowledge it to avoid another transmission. Typically, the ack of P_2 is transmitted if the medium is still idle after a timeout: it means P_1 is not currently transmitting an ack.

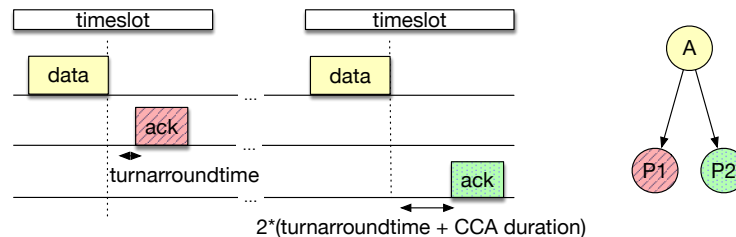


Fig. 1: Duocast Transmission (with two different parents)

We aim here to investigate which strategy to adopt to assign several receivers to the same cell (k-cast) to improve the reliability. With perfect links, k-cast transmissions are useless since the primary parent *always* receives the packet correctly. The other parents will have to awaken to receive the packet although they are useless. On the other hand, unreliable links may be combatted when several receivers can all decode the same packet.

Besides, opportunistic scheduling would be inefficient if the radio link *among* parents is very unreliable. In that case, a parent may not hear that a previous parent is currently transmitting an ack: it will start to transmit its own ack, which will create a collision, and retransmissions. Moreover, both parents are

convinced they both have to forward the packet: a *duplicated* packet has been created.

3.1 Number of cells to provision for a given set of parents

Let us first compute the number of cells a node has to reserve with its parent. We rely on RPL to construct the routes [15] (cf section 2.2). Then, the scheduling algorithm has to decide how many cells to reserve with its parent. We decide here to minimize the energy consumption, i.e. energy for reception and transmission so that at least one parent has received correctly the data packet.

Let denote by \mathcal{P} the set of parents selected by the node n to forward its packets. We compute the Packet Delivery Ratio (PDR) for this node, i.e. average number of transmissions before one parent has acknowledged correctly the packet. In most implementations, each node divides the number of packets it transmits by the number of acks it receives. When a packet is transmitted to several receivers / parents, the transmission has failed if none is able to send an ack:

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

where $PDR(p)$ denotes the average Packet Delivery Ratio toward the parent p , i.e. ratio of unicast packets received correctly by p .

We force the scheduling function to reserve a sufficient number of cells to achieve a given minimum reliability (e.g. $PDR_{th} = 99\%$) within the current slotframe. The number of transmission opportunities ($n_{txop}(n \rightarrow \mathcal{P})$) We denote by $n_{txop}(n \rightarrow \mathcal{P})$ the number of transmission opportunities to reserve to guarantee a minimum reliability. Typically, $n_{txop}(n \rightarrow \mathcal{P})$ must contain one copy and all its retransmissions to have a minimum Packet Delivery Ratio:

$$1 - (1 - PDR_{n \rightarrow \mathcal{P}})^{n_{txop}(n \rightarrow \mathcal{P})} \geq PDR_{th} \quad (2)$$

$$n_{txop}(n \rightarrow \mathcal{P}) = \frac{\log(PDR_{th})}{\log(1 - (1 - PDR_{n \rightarrow \mathcal{P}}))} \quad (3)$$

Finally, we have to provision enough cells to forward **all** the packets:

$$nbCells = n_{txop}(n \rightarrow \mathcal{P}) * n_{pkts}(n \rightarrow \mathcal{P}) \quad (4)$$

where $nbCells$ denotes the the number of cells reserve from n to its parents, and $n_{pkts}(n \rightarrow \mathcal{P})$ denotes the average number of packets that the node n has to forward to its parents within a slotframe.

Cells are then removed and inserted dynamically in the schedule to respect this reliability. To avoid oscillations, we use an hysteresis function, similarly to the scheduling function SFx [4].

3.2 Energy Consumption to choose the right set of parents

We should not allocate systematically all the parents to a given cell. Indeed, medium link qualities do not require to have so many receivers: only the first

parents will be actually useful, the other ones will waste their energy for idle listening. Inversely, allocating an insufficient number of receivers has a negative impact on the reliability and the delay: the source node needs more retransmissions to deliver the packet and to receive an **ack**.

Let us compute the energy consumption associated to a correct reception of a packet transmitted by a node n . For a sake of simplicity, a node assumes the link quality with its different parents may differ, but all the parents have paths with similar qualities toward the border router. \mathcal{P} keeps on denoting the set of parents selected by the node n to forward its packets.

Thus, the energy consumption (in transmission mode) is finally:

$$E_{tx}(n) = n_{txop}(n \rightarrow \mathcal{P}) * E_{tx} \quad (5)$$

where E_{tx} denotes the energy in TX mode for one cell.

Each parent has to be awake during all these cells. However, only some of them are actually busy, the other ones only correspond to idle listening. Thus, the energy to receive the packets of n can be estimated as:

$$E_{rx}(n) = n_{pkts}(n \rightarrow \mathcal{P}) * E_{rx} + (nbCells - n_{pkts}(n \rightarrow \mathcal{P})) * E_{idle} \quad (6)$$

where E_{rx} (resp. E_{idle}) denotes the energy in RX (resp. IDLE) mode for one cell.

We have consequently to compute the set of parents which minimizes the energy consumption

$$E_{total}(n) = E_{rx}(n) + E_{tx}(n) \quad (7)$$

Thus, the energy consumed by a node depends on the Packet Delivery Ratio toward its parents, and on the number of cells to provision to guarantee a given reliability. Inserting more parents increases the Packet Delivery Ratio and thus decreases the energy to transmit one packet. On the contrary, it also increases the energy consumed to receive the packets (idle and rx) for the parents. We will propose in the next section a strategy to decide which and how many parents to select.

3.3 Adaptive and Localized Scheduling Strategy

We propose here a scheduling algorithm adapted for k-cast. In particular, we have to decide how many and which parents should be allocated for each transmitting cell. Using a bad link instead of a good link is never relevant: a node should always use its best parents.

We propose a greedy approach to decide which cells to schedule (algo. 1). A node n applies the following approach:

1. n ranks its parents according to their PDR in descending order (line 2);
2. greedily, it inserts the next best parent in the list of forwarders (lines 5-6);
3. if the energy consumption (for both RX and TX) is reduced, then we continue the greedy allocation, and we consider the next parent. Else, the node n finally assigns the current list of forwarders (except the last parent) to its cells.

Algorithm 1: Parent Selection for Scheduling

Data: set of parents (P)
Result: set of forwarders (FW)
// initialization
1 $FW \leftarrow \emptyset$;
2 $\text{rank}(P, \text{PDR})$;
3 $\text{energy} \leftarrow \infty$;
// adds a new parent in the forwarder set
4 **repeat**
| // picks the first parent in the list
5 $p \leftarrow \text{popFirst}(P)$;
6 $FW \leftarrow FW \cup \{p\}$;
| // updates the energy cost according to eq. 7
7 $\text{energyOld} \leftarrow \text{energy}$;
8 $\text{energy} \leftarrow \text{computeEnergy}(FW)$;
| // repeat until the energy starts to increase
9 **until** $\text{energy} > \text{energyOld}$;
| // removes the last forwarder added in the list
10 $\text{removeLast}(FW)$;
11 **return** (FW);

In other words, the algorithm stops when a local minimum is achieved. To decide how many cells have to be allocated, the node n uses the eq. 4.

Theorem 1. *This greedy strategy allocates the set of parents which minimizes the energy consumption.*

Proof. Let us follow here a proof by contradiction. We order the parents by their decreasing PDR value. Our algorithm terminates with the set P . If the set P is not optimal, we replace one parent p of P by another parent p' with a smaller energy consumption. The reception energy of p is at least equal to those of p' since the number of cells with p' cannot be smaller (eq. 6) since p' provides a larger PDR than p .

We can prove similarly that when the algorithm discards the k^{th} parent because it starts to increase the energy consumption, another parent with a lower PDR cannot decrease further the energy consumption. ■

4 Performance Evaluation

We evaluate here the performance of IEEE802.15.4TSCH with k-cast relying on openwsn³. We re-use here the approach of SFx [4] to define how many cells have to be reserved. Each node computes dynamically the number of cells to use according to the number of packets to forward, and the Packet Delivery Ratio

³ <http://openwsn.org> provides an opensource implementation of IEEE802.15.4-TSCH

Parameter	Default value
Duration	60s
Traffic type	100 packets
Data packet size	127 bytes
Number of nodes	20 nodes
Number of receivers	2 parents (in the same slot)
Destination	1 sink (convergecast)
MAC layer	IEEE802.15.4-TSCH
Schedule	Distributed
Time slot duration	15ms
Slotframe length	101 slots
PHY model	fixed physical PDR for each radio link
physical PDR	uniformly picked between 20 and 90%

Table 1: Default values of the parameters in our simulations.

in the cells. Then, we decide how many parents have to be allocated to a given cell. Our implementation is fully available at <https://github.com/ineshos/6TSCH/tree/master/python/ScheduleSimulator/simulator>.

We rely here on simulations, and control the physical Packet Delivery Ratio (complement of the PER) of the links (Tab. 1). We assume the physical PDR remains fixed for the whole simulation. We first focus on a simple topology to evaluate the behavior of one flow, and then we study the performance in a random topology.

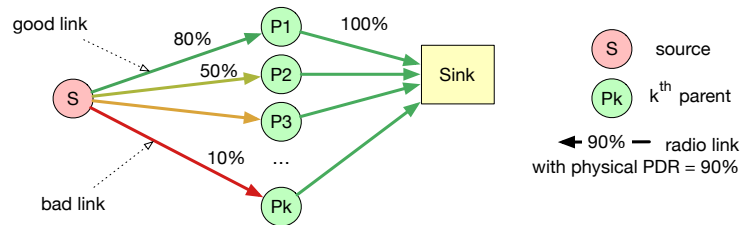


Fig. 2: Simple topology with multiple forwarding nodes with different link qualities.

4.1 Simple forwarding scenario

We first consider the topology illustrated in figure 2, where the physical Packet Delivery Ratio (i.e. the complement of the Packet Error Rate) has been fixed statically. The source generates 100 packets for the sink, 2 hops away.

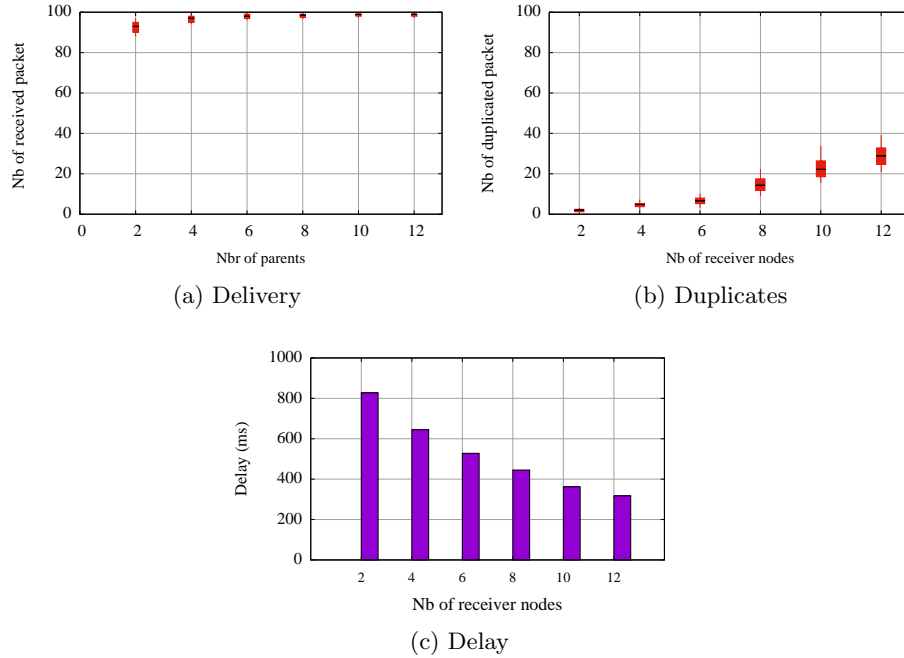


Fig. 3: Performance of a source with multiple parents with different link qualities.

Using more parents allows the network to increase the reliability: a single transmission may be received by any of the parents (Fig. 3a). However, using more than 6 parents seems to be inefficient. Indeed, the bad parents have only a marginal impact: they receive seldom the packets correctly. Besides, such parents would forward a packet only if all the previous parents (with a better link quality) have not transmitted an ack. This will occur very infrequently, and their impact on the reliability is very limited.

Figure 3b illustrates the number of duplicated packets (the packets which are received several times by the sink). With only two receivers, the amount of duplicates can be neglected (less than 2% of the packets). With many receivers, the number of duplicates becomes significant, and has a negative impact on the energy consumption: redundant and useless packets are forwarded by the different parents.

Figure 3c illustrates the delay. Because of the anycast transmission, we let *any* receiver forward the packet. Thus, the delay decreases with more receivers. Selecting many receivers has always a positive impact on the delay, even if it also increases the energy consumption.

Finally, we measured the number of cells that SFx reserves in the schedule (Fig. 4). The number of shared cells varies between 12 and only 3 cells for the whole network (20 devices), depending on the volume of traffic to forward. We remind that using dedicated cells requires to reserve at least one cell for each

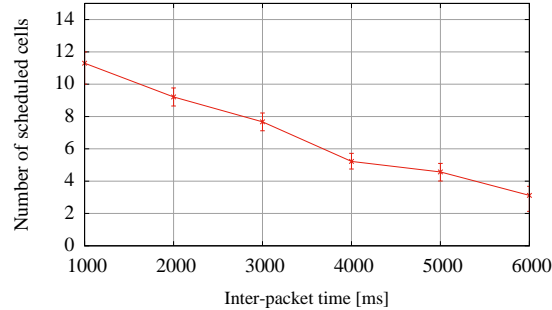


Fig. 4: Impact of the traffic load on the number of allocated cells.

device (thus 20 cells). Some additional cells would even be required to cope with retransmissions with non perfect radio links, which is the case here for most of the radio links. Thus, using shared cells is particularly efficient to reduce the number of allocated cells.

4.2 Random Topology

We now consider a random topology of 20 nodes, and assign a physical PDR to each radio link, uniformly distributed between 20 and 90%. This way, we verify that k-cast transmissions help to improve the efficiency even in more complex topologies.

We first measure the number of packets received when transmitting 100 packets (Fig. 5a). We make the same remarks as for the simple topology case: k-cast keeps on improving the reliability, to the same extent. Thus, the k-cast approach helps to improve globally the network-layer PDR: almost all packets are correctly delivered to the sink.

We measure then the amount of duplicated packets with a variable traffic (Fig. 5b). The amount of duplicated packets is linear with the inter-packet time. Thus, the solution is scalable and robust to the traffic conditions.

Finally, we consider the energy consumption in Figure 5c. Using more receivers decreases at the beginning the energy consumption since less retransmissions are required to deliver a packet. In particular, selecting 3 parents allows the network to reduce its energy consumption by almost 30% compared with the unicast approach. On the contrary too many receivers means that we increase idle listening, which has a negative impact on the energy consumption when the parent has a marginal contribution on the delivery (i.e. it is seldom involved in the transmission). We obtain the optimal energy efficiency with 5 parents. Obviously, this optimal number of receivers depends on the ability of the device to turn fast its radio off when a cell is unused. Thus, it depends on the radio chipset specifications.

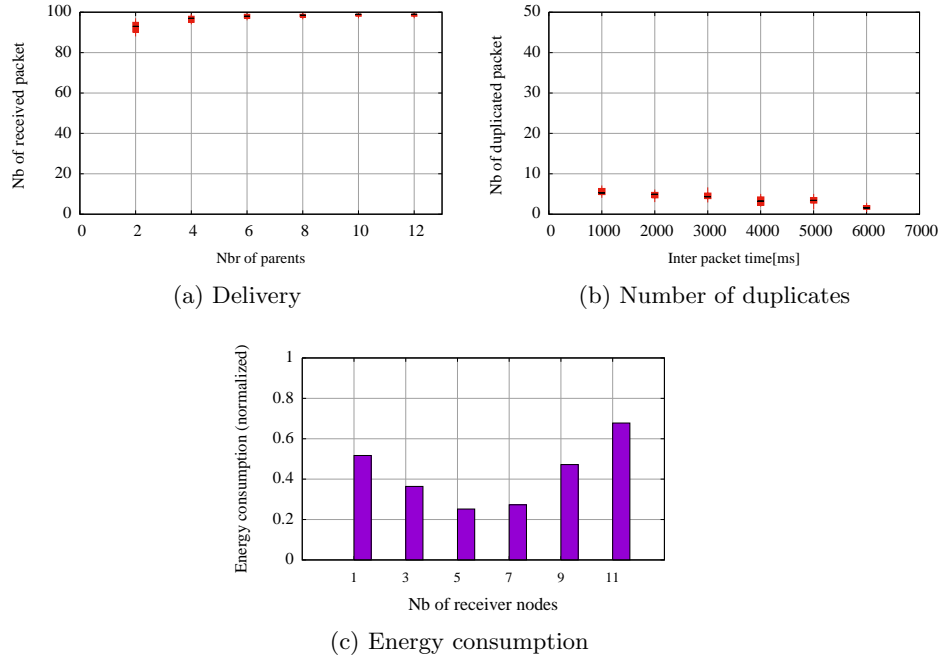


Fig. 5: Random topology – 20 nodes, physical PDR picked uniformly between 10 and 90%.

5 Conclusion & Perspectives

We propose a distributed algorithm to exploit k-cast scheduling: several receivers are assigned to the same transmission, to improve the probability of success. Our algorithm computes the set of parents to schedule so that the energy consumption is minimized, considering both the RX and TX mode. Our simulations show k-cast helps to improve the reliability with a low energy consumption. The amount of duplicated packets is reasonable when a small number of parents (e.g. 2 or 3) are assigned to the same cell.

In the future, we plan to investigate the actual cost of implementing such k-cast feature in TSCH. In particular, a larger number of `acks` means often a longer timeslot, which may waste energy. Besides, we also plan to investigate the number of duplicated packets which may be generated by complex topologies (hidden terminals, asymmetrical links, etc.)

References

1. IEEE Standard for Low-Rate Wireless Personal Area Networks (LR-WPANs). IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011) (April 2016). <https://doi.org/10.1109/IEEESTD.2016.7460875>

2. Al-Anbagi, I., Erol-Kantarci, M., Mouftah, H.T.: A Survey on Cross-Layer Quality-of-Service Approaches in WSNs for Delay and Reliability-Aware Applications. *IEEE Communications Surveys Tutorials* **18**(1), 525–552 (Firstquarter 2016). <https://doi.org/10.1109/COMST.2014.2363950>
3. Dobsław, F., Zhang, T., Gidlund, M.: End-to-End Reliability-aware Scheduling for Wireless Sensor Networks. *IEEE Transactions on Industrial Informatics* **12**(2), 758–767 (Dec 2014). <https://doi.org/10.1109/TII.2014.2382335>
4. Dujovne, D., Grieco, L.A., Palattella, M.R., Accettura, N.: 6TiSCH Experimental Scheduling Function (SFX). draft 1, IETF (March 2018)
5. Hashimoto, M., Wakamiya, N., Murata, M., Kawamoto, Y., Fukui, K.: End-to-end reliability- and delay-aware scheduling with slot sharing for wireless sensor networks. In: COMSNETS. pp. 1–8 (Jan 2016). <https://doi.org/10.1109/COMSNETS.2016.7439984>
6. Hermeto, R.T., Gallais, A., Theoleyre, F.: 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). <https://doi.org/10.1016/j.comcom.2017.10.004>
7. Huynh, T., Theoleyre, F., Hwang, W.J.: On the interest of opportunistic anycast scheduling for wireless low power lossy networks. *Computer Communications* **104**, 55 – 66 (2017). <https://doi.org/10.1016/j.comcom.2016.06.001>
8. Iova, O., Theoleyre, F., Noel, T.: Using multiparent routing in RPL to increase the stability and the lifetime of the network. *Ad Hoc Networks* **29**, 45 – 62 (2015). <https://doi.org/10.1016/j.adhoc.2015.01.020>
9. ISA-100.11a-2011:: Wireless systems for industrial automation: process control and related applications. *International Society of Automation (ISA) Std.* **1** (May 2011)
10. Kafi, M.A., Othman, J.B., Badache, N.: A Survey on Reliability Protocols in Wireless Sensor Networks. *ACM Comput. Surv.* **50**(2), 31:1–31:47 (May 2017). <https://doi.org/10.1145/3064004>
11. Li, X., Li, D., Wan, J., Vasilakos, A.V., Lai, C.F., Wang, S.: A review of industrial wireless networks in the context of Industry 4.0. *Wireless Networks* **23**(1), 23–41 (Jan 2017). <https://doi.org/10.1007/s11276-015-1133-7>
12. Palattella, M.R., Accettura, N., Grieco, L.A., Boggia, G., Dohler, M., Engel, T.: On Optimal Scheduling in Duty-Cycled Industrial IoT Applications Using IEEE802.15.4e TSCH. *IEEE Sensors Journal* **13**(10), 3655–3666 (Oct 2013). <https://doi.org/10.1109/JSEN.2013.2266417>
13. Papadopoulos, G., Matsui, T., Thubert, P., Watteyne, T., Montavont, N., Texier, G.: Leapfrog Collaboration: Toward Determinism and Predictability in Industrial-IoT applications. In: ICC. pp. 1–6. IEEE (2017). <https://doi.org/10.1109/ICC.2017.7997160>
14. Q. Wang, et al.: 6top protocol (6p). draft 12, IETF (June 2018)
15. T. Winter, et al. : Rpl: Ipv6 routing protocol for low-power and lossy networks. rfc 6550, IETF (2012). <https://doi.org/10.17487/RFC6550>
16. Theoleyre, F., Papadopoulos, G.Z.: Experimental Validation of a Distributed Self-Configured 6TiSCH with Traffic Isolation in Low Power Lossy Networks. In: MSWiM. pp. 102–110. ACM (2016). <https://doi.org/10.1145/2988287.2989133>
17. Wang, X., Wu, X., Zhang, X.: Optimizing opportunistic routing in asynchronous wireless sensor networks. *IEEE Communications Letters* **21**(10), 2302–2305 (Oct 2017). <https://doi.org/10.1109/LCOMM.2017.2729557>
18. Watteyne, T., Mehta, A., Pister, K.: Reliability through frequency diversity: Why channel hopping makes sense. In: PE-WASUN. pp. 116–123. ACM (2009). <https://doi.org/10.1145/1641876.1641898>