

# High-Reliability Scheduling in Deterministic Wireless Multi-hop Networks

Guillaume Gaillard, Dominique Barthel, Fabrice Theoleyre, Fabrice Valois

► **To cite this version:**

Guillaume Gaillard, Dominique Barthel, Fabrice Theoleyre, Fabrice Valois. High-Reliability Scheduling in Deterministic Wireless Multi-hop Networks. PIMRC 2016 - IEEE 27th Annual International Symposium on Personal, Indoor and Mobile Radio Communications, Sep 2016, Valencia, Spain. <<http://www.ieee-pimrc.org/geninfo.html>>. <hal-01344792>

**HAL Id: hal-01344792**

**<https://hal.inria.fr/hal-01344792>**

Submitted on 12 Jul 2016

**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.

# High-Reliability Scheduling in Deterministic Wireless Multi-hop Networks

Guillaume Gaillard<sup>3</sup>, Dominique Barthel<sup>1</sup>, Fabrice Theoleyre<sup>2</sup>, and Fabrice Valois<sup>3</sup>

<sup>1</sup>Orange Labs R&D, F-38240, Meylan, France. dominique.barthel@orange.com

<sup>2</sup>ICube, Université de Strasbourg / CNRS, F-67412 Illkirch, France. theoleyre@unistra.fr

<sup>3</sup>Univ Lyon, INSA Lyon, Inria, CITI, F-69621 Villeurbanne, France.  
{fabrice.valois,guillaume.gaillard1}@insa-lyon.fr

July 12, 2016

## Abstract

The industrial Internet of Things (IoT) relies on multi-hop radio paths. Synchronized nodes follow a Frequency-Time Division Multiple Access (FTDMA) schedule, but even using channel-hopping to mitigate interference, the radio links suffer packet losses. Resource allocation algorithms must consider the requirements of the applications in terms of delivery and allocate extra resource to compensate for anticipated losses.

We propose a hop-by-hop allocation mechanism that extends the Traffic-Aware Scheduling Algorithm (TASA) by enabling retransmissions. We give each flow on the network the possibility to satisfy its applicative end-to-end delivery constraint. We keep the amount of resource allocated for retransmissions low, and balance the allocations on the relay nodes. By means of simulations, we show the gain in terms of reliability, and the cost in terms of number of allocations.<sup>1</sup>

## 1 Introduction

In the Internet of Things (IoT), the radio communications between sensor nodes are subject to col-

lisions, fading, interference, that negatively impact the Packet Error Rate (PER) [4]. When addressing the delivery of different client flows, the reliability is challenging. The end-to-end Packet Delivery Ratio (PDR) of a flow depends on the PER of each hop in the path to a gateway.

We claim that network operators must consider the traffic requirements, the radio topology, and the quality of the links to organize the communications. The state-of-the-art Traffic-Aware Scheduling Algorithm (TASA) enables to centrally compute a FTDMA schedule [9]. The authors claim optimality in terms of schedule compactness. The TASA scheduler focuses on the traffic each node locally generates. It is a relevant centralized solution for scheduling in 6TiSCH networks [10].

However, TASA does not take into account the reliability concerns: the end-to-end packet delivery ratio (PDR) is not the priority. In particular, TASA does not consider:

- the robustness to packet losses;
- the adaptability to an increase in traffic;
- the fragmentation of long packets: the success of delivery depends on all the fragments of each message.

<sup>1</sup>Accepted for presentation at IEEE PIMRC 2016.

In this work, we go beyond TASA by including robustness to the schedules. We propose to allocate additional resource for frame retransmissions to increase the probability of success of a delivery. It enables to obtain a given minimum PDR, but it impacts the energy consumption by increasing the quantity of allocated resource. We also modify the load-balancing when adding retransmission opportunities.

We provide a hop-by-hop mechanism that maintains the PDR along each path, and chooses the number of time-frequency blocks (namely cells) added to the schedule for the retransmissions. We adapt these allocations counts to the PER of each link and to the allocation load on the nodes. We integrate the mechanism in TASA and compare the results through simulations.

We extend TASA and our contribution is three-fold:

1. we provide an algorithm that computes the necessary number of additional resource to satisfy the expected end-to-end PDR. Our algorithm distributes hop-by-hop retransmission cells, and balances the load over each link while reducing the overhead;
2. we implement an over-provisioning algorithm for multiple flows with fragmentation;
3. we compare the performance of TASA and our proposal in terms of PDR satisfaction, and cost in allocations.

The paper is organized as follows: we detail the related work (§2) and describe our model (§3). Then, we detail our contribution in section 4; we finally evaluate the performance of our solution (§5) and conclude (§6).

## 2 Related work

IEEE802.15.4e-TSCH represents a key medium access protocol for industrial wireless networks requiring high reliability [2]. Slow channel hopping improves the reliability in noisy environments and a TDMA scheme is adopted to make the performance deterministic. A centralized scheduler may compute

the transmissions opportunities for each radio link while avoiding the collisions.

The IETF 6TiSCH Working Group aims at providing the mechanisms to operate an IPv6 network on top of IEEE802.15.4e-TSCH [10]. Some mechanisms are in particular proposed to configure the schedule of the nodes. Besides, 6TiSCH ensures traffic isolation by exploiting the concept of *track*: bandwidth is reserved for a particular flow, from the source to the destination. Thus, an application/flow is not impacted by another application, which uses orthogonal resource (i.e. different channel offsets or time slots).

With the Traffic-Aware Scheduling Algorithm (denoted TASA) [8], Palattella *et al.* provided a pioneering piece of work to define a centralized scheduling algorithm with IEEE 802.15.4e-TSCH. The algorithm aims at minimizing the schedule length, improving its compactness. The priority is given to the nodes with the largest number of packets to transmit. Besides, TASA schedules concurrent (*duplex-conflict free*) transmissions over orthogonal channels. However, TASA does not consider lossy links.

DETAS [3] proposes a decentralized scheduling solution. The algorithm alternates the transmitting and receiving time slots. In other words, a node which receives a frame, will forward it in the next consecutive time slot. Thus, if a packet is lost, the rest of the bandwidth is wasted along the path.

Dobslaw *et al.* propose SchedEx [5], to make a schedule reliable. The authors calculate the number of necessary retransmissions for all the packets, at each link of the routing tree. This expected number of retransmissions is defined according to the load of a radio link and its reliability. In other words, Schedex does not guarantee flow isolation with differentiated PDR requirements.

Yang *et al.* [12] constructed an optimal schedule for time-sensitive flows: new cells are inserted in the schedule if the end-to-end reliability is insufficient and until the deadline constraint is not fulfilled anymore. Yigit *et al.* [13] studied the impact of routing on the scheduling: using unreliable links increases the number of time slots required to achieve a minimum reliability.

In this paper, we propose to take into account both fragmentation and packet loss. We present a schedul-

ing algorithm which guarantees a minimum end-to-end reliability for each flow, allocating an accurate number of cells for each fragment and its retransmissions.

### 3 Model

In order to construct reliable FTDMA schedules, we model the behavior of the network for both the scheduler and the over-provisioning algorithm. The parameters of the model are summarized in Table 1.

#### 3.1 Topology Model

The sensor network is a set of nodes exchanging messages in a multi-hop way. 3 distinct types of nodes are considered:

- the leaf nodes which only generate traffic;
- the intermediary relays that forward it;
- one or several gateways that collect it.

The nodes have a single half-duplex radio interface and have buffering capacities. Each leaf node carries one or several applications that generate messages of constant size (one or several frames). For each application, a leaf node creates one flow of constant-size messages directed to a gateway.

We consider the Packet Error Rate (PER) to characterize the link quality. The PER represents the ratio of unsuccessful transmissions of frames on a link. Both classical propagation models and empirical results give an estimation of the relationship between the PER and the distance, attenuation, and type of node [7]. The reality often differs from the model: a PER depends on other parameters such as interference, noise, state of the nodes [4]. We assume that a monitoring mechanism provides frequent updates on the link qualities that make possible the schedule adaptation. We adopt the simplified path-loss model presented in [7].

We consider that the PER is time-invariant on the scale of the scheduling, and independent of the channel offset.

Table 1: Parameters for the model

Variable	Explication
$f \in F$	A flow $f$ among the set of flows $F$
$path(f)$	The path of $f$
$l_1, l_2, \dots, l_{gw}$	The set of links of the path of $f$
$n_{msg}(f)$	Messages in a slotframe from $f$
$n_{frag}(f)$	Fragments in a message from $f$
$PDR_{msg}^{min}(f)$	Min. message end-to-end PDR for $f$
$PDR_{msg}(f)$	Message end-to-end PDR for $f$
$PDR_{msg}^{hop}(f, l)$	PDR of the messages from $f$ at link $l$
$per(l)$	Packet Error Rate (PER) on a given link $l$
$n_{cell}(l)$	Number of allocated cells on a given link $l$
$load(l)$	Allocation load on a given link $l$
$alloc_f$	Set of allocation counts along $path(f)$
$alloc_f^{i_{th}}$	Allocation count of the $i_{th}$ link of $path(f)$
$track_f$	Cells constituting the track of $f$ along $path(f)$
$track_f^{i_{th}}$	Cells allocated for $f$ at the $i_{th}$ link of $path(f)$
$n_{rtx}^{max}(f)$	Max. num. of retransmissions (hop, message) for $f$
$Sol(f)$	The set of solutions of the algorithm

#### 3.2 Communications and resource allocation

The FTDMA schedule is a matrix of time slots (of constant duration) and frequency channels (e.g. 16). One allocation is an assignment of a time-frequency block, named *cell*, to the transmission of one fragment between two neighboring nodes. Each message is divided into one or several fragments that separately transit on a single cell each (Fig 1).

We isolate each flow by allocating them different cells. The set of cells of a given flow is named its *track*.

We consider a centralized scheduling algorithm that takes as input two topological parameters:

1. a routing acyclic graph. For each flow, we build a loop-free path from the leaf node to a gateway. In the case of TASA, the routing graph uses RPL information [11]. RPL is a distributed routing algorithm that lets each relay select the parent that minimizes the cumulated ETX metric to the root. Our model is also applicable to multi-path routing.

Our algorithm computes the number of cells of a track at each link, for a given flow. The final track allocation is the sum of the allocations for each constant-size message.

2. the Packet Error Rate (PER) of every link. A

set of links interfere if there is a loss of information when transmissions take place at the same time slot and channel. We need to build the conflict graph, i.e. the information about the set of interfering links.

In case no other information is given about interference, we build the conflict graph by considering that the three-hop neighbors and beyond are not interferers.

Each application has its own reliability constraint, expressed as the expected applicative end-to-end Packet Delivery Ratio (PDR), denoted  $PDR_{msg}^{min}(f)$ . For each flow  $f$ , the PDR is the ratio between the number of messages received at the gateway, and the number of messages sent from the source. A schedule satisfies the reliability constraint for a given flow if the minimum PDR is met when computed over an agreed-upon fixed time window (e.g one minute/hour/day).

### 3.3 Traffic definition

We consider an heterogeneous traffic, mixing several applications [1]. We assume that the traffic of each application is bounded by a periodical amount of messages. We address the allocation of resource for periodical traffic patterns. The FTDMA schedule is hence divided into periodical slotframes (set of time slots) that repeat in time, indefinitely.

From the perspective of the allocation algorithm, all the nodes have buffered the fragments of the messages at the beginning of the slotframe. The fragments must have reached a gateway during the same slotframe.

Each node is capable of detecting the loss of a fragment (if it does not receive an acknowledgement). In this case, it retransmits the fragment to the next-hop neighbor during the next cell associated with the same track ID. The schedule algorithm limits the number of hop-by-hop retransmissions for a message to a maximum per hop (e.g. 15 fragments). This maximum applies for each message. This way, we avoid the preemption of all the resource by a given track.

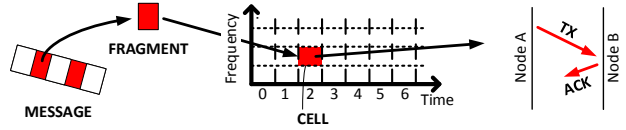


Figure 1: The fragments of a message

Fig 1 shows how the fragments of a message transit on a scheduled cell, between two nodes A and B. The cell comprises the transmission of a fragment and its acknowledgement.

## 4 The hop-by-hop over-provisioning

We now provide an algorithm to compute, for each flow, the number of cells needed at each link of the path, for the transmissions and retransmissions of the fragments of each message. These communication opportunities are dimensioned so that the PDR requirements of the flow are addressed while limiting the number of overhead cells.

For each flow  $f$ , we must meet the reliability constraint, expressed as the satisfaction of the end-to-end PDR:

$$PDR_{msg}(f) \geq PDR_{msg}^{min}(f) \quad (1)$$

Our objective consists in balancing the load among the links. Our approach reduces the total number of cell allocations and minimizes the maximum allocation load over the path.

We address the load-balancing objective at the level of each flow. Indeed, we assume that the messages forming the same flow have a constant size. Thus, an allocation message-by-message would increase the complexity and latency of the algorithm uselessly.

In order to reduce the differences between the allocation loads of the links, we want to minimize the maximum link load over each path:

$$\min_{alloc_f \in Sol(f)} \left( \max_{l \in path(f)} load(l) \right) \quad (2)$$

## 4.1 Expression of the end-to-end PDR with retransmissions

We consider independent packet loss probability. On each hop, the successful transmission of a message depends on the successful transmissions of its fragments. It is necessary that at least one success occurs for each fragment of the message, all of them within the available opportunities (the allocation count). In other words, we need at least  $n_{frag}(f)$  successes among  $alloc_f^l$  intents (cf. Table 1 for the notations). This condition is expressed by a partial cumulative distribution function of the binomial formula:

$$\forall f \in F, \forall l \in path(f), \text{ with } n = alloc_f^l, p = per(l)$$

$$PDR_{msg}^{hop}(f, l) = \sum_{k=0}^{n-n_{frag}(f)} \binom{n}{k} p^k (1-p)^{n-k} \quad (3)$$

For the link  $l \in path(f)$ ,  $alloc_f^l$  is the allocation count,  $per(l)$  is the Packet Error Rate,  $PDR_{msg}^{hop}(f, l)$  is the PDR for one message of  $f$ .

Considering each hop independent from the others, our constraint (1) can be expressed as the product along the path of each hop expression:

$$PDR_{msg}(f) = \prod_{l \in path(f)} PDR_{msg}^{hop}(f, l) \quad (4)$$

## 4.2 Satisfaction of the reliability constraint with balanced hop-by-hop retransmissions

The algorithm has to find a set of allocation counts satisfying the constraint (1). We denote  $alloc_f$  the set of allocation counts along the path, for a message of flow  $f$ :

$$alloc_f = \left\{ alloc_f^i, l_i \in [l_1 : l_{gw}] \right\} \quad (5)$$

with the links  $[l_1; l_{gw}]$  constituting the path of the flow  $f$ .

For each link of the path, the track of  $f$  is defined by the sum of the allocation count of each message:

$$track_f = n_{msg}(f) \cdot alloc_f$$

$$\forall l \in path(f), track_f^l = n_{msg}(f) \cdot alloc_f^l \quad (6)$$

At each link, a message must obtain at least enough allocations for all its fragments and at most the maximum allowed number of retransmissions:

$$\forall l \in path(f), n_{frag}(f) \leq alloc_f^l \leq n_{rtx}^{max}(f) + n_{frag}(f) \quad (7)$$

We denote as  $Sol(f)$  the set of solutions of the algorithm. Given the assumption that the number of retransmissions for the fragments of each message is limited to a maximum (Eq. (7)), we want to distribute the allocations on the links network-wide. The allocation load is the sum of the already allocated cells for a given link, and the cells allocated for  $f$ :

$$\forall l \in path(f), load(l) = (n_{cell}(l) + n_{msg}(f) \cdot alloc_f^l) \quad (8)$$

Since we aim at minimizing the maximum load, we propose an inverse greedy algorithm that gives a solution to this problem, flow by flow.

## 4.3 Inverse greedy algorithm for over-provisioning

At the beginning, and for each flow  $f$ , the over-provisioning algorithm (Alg. 1) initializes the allocation counts for each message at each hop at its maximum value (line 2). If the minimum end-to-end PDR is not satisfied (Eq. (1)), no solution exists. The flow is discarded (cf. condition line 5) and the algorithm proceeds with another flow.

Otherwise, the algorithm repeats over the following 3 steps:

1. We find the most loaded link of the path, among the links that have not yet been treated (line 8);
2. We decrement its allocation count and update the load for this link (line 9);
3. We verify that the end-to-end PDR constraint (Eq. (1)) is still satisfied and that the transmission of all the fragments is still possible (Eq. (7)) (line 12).

If this condition does not hold, we update the variables to the last valid value (line 13), and set the considered link as treated (line 16).

---

**Algorithm 1:** Finding the allocation counts for a given flow  $f$ 


---

**Data:**  $f \in F, path(f), n_{msg}(f), n_{frag}(f), n_{rtx}^{max}(f)$   
 $\forall l \in path(f), n_{cell}(l), PDR_{msg}^{min}(f),$

**Result:**  $\forall l \in path(f), load(l),$   
 $alloc_f \in Sol(f)$  satisfying Eq. (2) and Eq. (1)

```

1  $treated \leftarrow \emptyset;$ 
2  $\forall l \in path(f), alloc_f^l \leftarrow n_{frag}(f) + n_{rtx}^{max}(f);$ 
3  $\forall l \in path(f), load(l) \leftarrow (n_{cell}(l) + n_{msg}(f) \cdot alloc_f^l);$ 
4 update  $PDR_{msg}(f)$  applying Eq. (4);
5 if  $PDR_{msg}(f) \geq PDR_{msg}^{min}(f)$  then
6   while  $\exists l \in path(f), l \notin treated$  do
7     repeat
8       Find one  $l_{max}(f) \in path(f) \setminus treated$  such
       that  $load(l_{max}(f)) =$ 
        $\max_{l \in path(f), l \notin treated} (load(l));$ 
9        $alloc_f^{l_{max}(f)} \leftarrow alloc_f^{l_{max}(f)} - 1;$ 
10      update  $PDR_{msg}(f)$  applying Eq. (4);
11      update  $load(l_{max}(f))$  applying Eq. (8);
12    until  $PDR_{msg}(f) < PDR_{msg}^{min}(f)$  or
     $alloc_f^{l_{max}(f)} < n_{frag}(f);$ 
13     $alloc_f^{l_{max}(f)} \leftarrow alloc_f^{l_{max}(f)} + 1;$ 
14    update  $PDR_{msg}(f)$  applying Eq. (4);
15    update  $load(l_{max}(f))$  applying Eq. (8);
16     $treated \leftarrow treated \cup \{l_{max}(f)\};$ 
17  end
18   $\forall l \in path(f), n_{cell}(l) \leftarrow load(l)$ 
19 end

```

---

Note that the algorithm will iteratively reduce the load of the most loaded link (step 1). This link may be different for each iteration, until the PDR constraint does not hold anymore: we cannot reduce anymore the load of this link.

When the allocation counts have been successively computed for all the links of the path, we update the number of allocated cells for each node with the new values (line 18) and we consider the next flow.

Fig 2 illustrates an iteration of the algorithm for one path of 3 links, and a flow with one message (3 fragments). Link  $L_2$  has already been treated. For the considered flow, the algorithm did not provision retransmissions since it is the most loaded link, due to the allocations of the previous flows ( $n_{cell}(L_2)$  is the highest of the 3 links). Thus the algorithm, when trying to reduce the maximum load, has first reached its minimum value by decrementing the retransmission

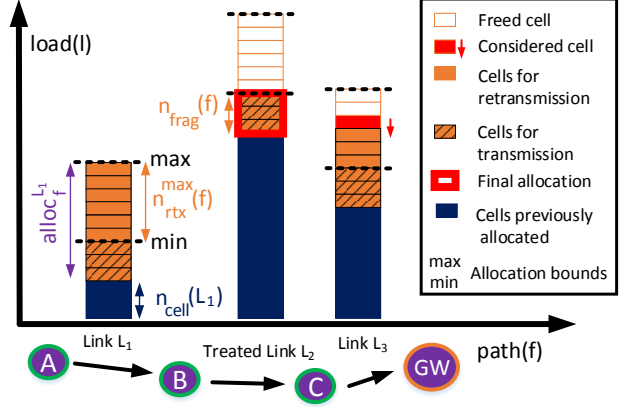


Figure 2: Example of a given iteration of the algorithm for a flow of 1 message

opportunities. Link  $L_1$  has not been treated yet. Its allocation number is maximum. As the most loaded untreated link,  $L_3$  is decremented one cell.

We find for each flow the set of allocation counts with minimal maximum value. We prove this minimality in [6].

#### 4.4 Extending TASA to over-provisioning

Because we consider flow-level retransmissions, we slightly modify the original TASA algorithm. We adapt its set up by considering each flow separately. We fill the queues of the leaf nodes with the initial fragments. We differentiate the cells allocated for a transmission with the ones for a retransmission.

At each slot of the allocation, TASA first sums the remaining upward traffic that has to be allocated through each node in the network. We add the retransmission cells to each sum.

TASA selects a set of nodes, having non-empty queue, to transmit a fragment at the same time slot. We consider that the nodes that still have unscheduled retransmission cells also have non-empty queue. These cells have the same priority as the transmission cells.

In the queue of a node, the original TASA applies a classical first in, first out (FIFO) method. We modify

this approach by choosing to schedule, message after message, first the oldest fragment transmissions, then the corresponding retransmissions.

The queues are updated at the end of the allocation of each message. Thus, a node receives all the retransmissions (and fragments) before forwarding a message. This enables a route-over mechanism, because they provide consecutive cells that give every node the possibility to collect all the fragments before forwarding them again.

More details on the integration of retransmission opportunities in TASA can be found in [6].

## 5 Evaluation

In this section, we compare the original TASA [8] with our enhanced version of the algorithm, denoted as  $TASA_{rtx}$ , including TASA with hop-by-hop retransmission capability. We show under various conditions that this extension schedules reliable communications while balancing the network load.

### 5.1 Scenario

We run Monte Carlo simulations (using a Python script) on a given set of topologies, by varying the default parameters (Table 2).

We define two applications (app. 1 and app. 2) running on half the leaf nodes each. Their PDR constraint incrementally varies between the default value (Table 2) and the maximum value (1.0). The variations are expressed as a percentage of increment in the figures.

The leaf nodes are uniformly spread in a rectangle of  $400 \times 200$  meters. The relays are placed on a triangular mesh (every approximately 70 meters). The 2 gateways are placed at central positions (Table 2).

We model the PER of each link according to a path-loss propagation model [7]. We choose the path-loss exponent values, the transmission power, and the reference distance (Table 2) depending on the type of node and reflecting their environment [6].

Based on the PER values, we construct a routing tree rooted at each gateway, according to the ETX

Table 2: Parameters of the simulation.

Parameter	value	range in simulations	
Size of slotframe (slots)	1000	100 to 1000	
Rectangle dimensions		400m $\times$ 200m	
Number of channels	16		
Max. number of retransmissions	16		
Leaf nodes	200	uniform distribution	
Relay nodes	24	triangle mesh, every 70m	
Gateways	2	(100m,100m),(300m,100m)	
Messages per slotframe, app. 1	1	1 to 10	
Messages per slotframe, app. 2	1	1 to 10	
Fragments per message, app. 1	3		
Fragments per message, app. 2	2		
Min. end-to-end PDR, app. 1	0.97	0.97 to 0.997	
Min end-to-end PDR, app. 2	0.80	0.80 to 0.98	
Propagation model	leaf-relay	relay-relay	relay-gateway
Path-loss exponent	3.5	2.5	1.9
Transmission power (dBm)	0	3	3
Reference distance (m)	10	22	22

metric. We build the conflict graph based on the 2-hops neighborhood.

We compare TASA and our  $TASA_{rtx}$  over 10 randomly generated topologies for 10 values of the given parameters:

1. The **traffic intensity**: we increase the number of messages for each flow;
2. The **slotframe size**: when the schedules are too long to fit in the slotframe, only a portion of them is applied;
3. The **expected PDR constraint** itself: we make the PDR satisfaction harder.

We evaluate the performance of our approach according to two criteria:

1. **satisfaction ratio**: the ratio of flows which meet the PDR constraint;
2. **maximum allocation load**: the maximum number of allocations by node and slotframe.

### 5.2 Results

The boxplots (Fig. 3) represent the minimum, maximum and the quartiles of the values obtained for the 10 topologies. The line binding the boxplots represents the mean of these values.



Fig. 3 shows that the original scheduler (TASA) does not permit to satisfy the reliability constraint on lossy links. The over-provisioning mechanism enables the satisfaction of PDR constraints, when the network limit of capacity is not yet reached. The original TASA does not take into consideration the PDR constraint. In our simulation, around 40 flows (20%) luckily satisfy their PDR constraint without any retransmissions. In Fig. 3a, this proportion decreases from 20% to 10% when the constraint increases. On the contrary, our algorithm enables to meet the PDR constraint for the messages of more than 95% of the flows.

The PDR is degraded when the slotframe is too short to contain all the required cells (less than 300 slots for TASA, 500 slots for  $TASA_{rtx}$ ) (Fig. 3c). The schedule is indeed larger with retransmissions opportunities.

In Fig. 3b, the number of cells allocated to the most loaded relay node, for transmission or reception, is constant for TASA: TASA does not consider the PDR constraint. With the hop-by-hop mechanism, the schedule remains acceptable (the maximum load on a given node is generally less than half the slotframe size (1000 slots): there is remaining capacity). The number of retransmission cells slowly increases for low values of the PDR constraint. Then, for the values of 60% of increment and beyond, the satisfaction of the constraint requires more cells and the cost in allocations rapidly increases.

We obtain a mean maximum node load of 270 allocated cells for large slotframe sizes (from 500 to 1000 slots in Fig. 3d). The over-provisioning does not drive the network to saturation.

The network is rapidly saturated (the most loaded relay is occupied at 100%) under the influence of the traffic load (Fig. 3f). With high traffic intensity (4 messages per leaf) the schedule length reaches the slotframe size (1000 slots). From this point the increase in the number of allocations for the most loaded node is less important (the two algorithms converge) because the schedule is limited to the slotframe size.

For both TASA and  $TASA_{rtx}$  the satisfaction of the PDR constraint decreases (Fig. 3e), but less rapidly with retransmission opportunities (because

they are prioritized for each flow in the queues, at the expense of the fragments of the other flows).

## 6 Conclusion

In a network where multiple applications have different delivery constraints, the operator must offer differentiated Quality of Service (QoS) in terms of reliability.

In this work, we provide an efficient way to give reliability to FTDMA schedules. We propose an over-provisioning mechanism that adapts to a requested end-to-end packet delivery ratio, taking as input parameter the quality of each link in a path. We extend the TASA scheduler by allocating resources for retransmissions.

We implement and compare our solution with the original TASA. Our results show that we enhance the scheduling algorithm by obtaining the requested reliability for 95% of the traffic, while keeping free more than half of the remaining capacity.

## References

- [1] Spectrum Requirements for Short Range Device, Metropolitan Mesh Machine Networks (M3N) and Smart Metering (SM) applications. *ETSI TC ERM, TR 103 055, v1.1.1*, Sept. 2011.
- [2] IEEE Standard for Local and metropolitan area networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer. *IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011)*, pages 1–225, 2012.
- [3] N. Accettura, M. R. Palattella, G. Boggia, L. A. Grieco, and M. Dohler. Decentralized traffic aware scheduling for multi-hop low power lossy networks in the internet of things. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a*, pages 1–6. IEEE, 2013.

- [4] 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*, 8(4):1–33, Sept. 2012.
- [5] F. Dobsław, T. Zhang, and M. Gidlund. End-to-end reliability-aware scheduling for wireless sensor networks. *Industrial Informatics, IEEE Transactions on*, PP(99):1–1, 2014.
- [6] G. Gaillard, D. Barthel, F. Theoleyre, and F. Valois. Enabling Flow-level Reliability on FT-DMA Schedules with efficient Hop-by-hop Overprovisioning. Research Report RR-8866, INRIA Grenoble Rhône-Alpes, 2016.
- [7] A. Goldsmith. *Wireless communications*. Cambridge university press, 2005.
- [8] M. R. Palattella, N. Accettura, M. Dohler, L. A. Grieco, and G. Boggia. Traffic aware scheduling algorithm for reliable low-power multi-hop ieee 802.15.4e networks. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*, pages 327–332, 2012.
- [9] M. R. Palattella, N. Accettura, L. A. Grieco, G. Boggia, M. Dohler, and T. Engel. On optimal scheduling in duty-cycled industrial iot applications using ieee802.15.4 e tsch. *Sensors Journal, IEEE*, 13(10):3655–3666, 2013.
- [10] P. Thubert, T. Watteyne, M. Palattella, X. Vilajosana, and Q. Wang. Ietf 6tsch: Combining ipv6 connectivity with industrial performance. In *IMIS*, pages 541–546, July 2013.
- [11] T. Winter, P. Thubert, and al. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550, IETF, March 2012.
- [12] M. Yan, K.-Y. Lam, S. Han, E. Chan, Q. Chen, P. Fan, D. Chen, and M. Nixon. Hypergraph-based data link layer scheduling for reliable packet delivery in wireless sensing and control networks with end-to-end delay constraints. *Information Sciences*, 278:34 – 55, 2014.
- [13] M. Yigit, O. D. Incel, and V. C. Gungor. On the interdependency between multi-channel scheduling and tree-based routing for WSNs in smart grid environments. *Computer Networks*, 65(0):1 – 20, 2014.

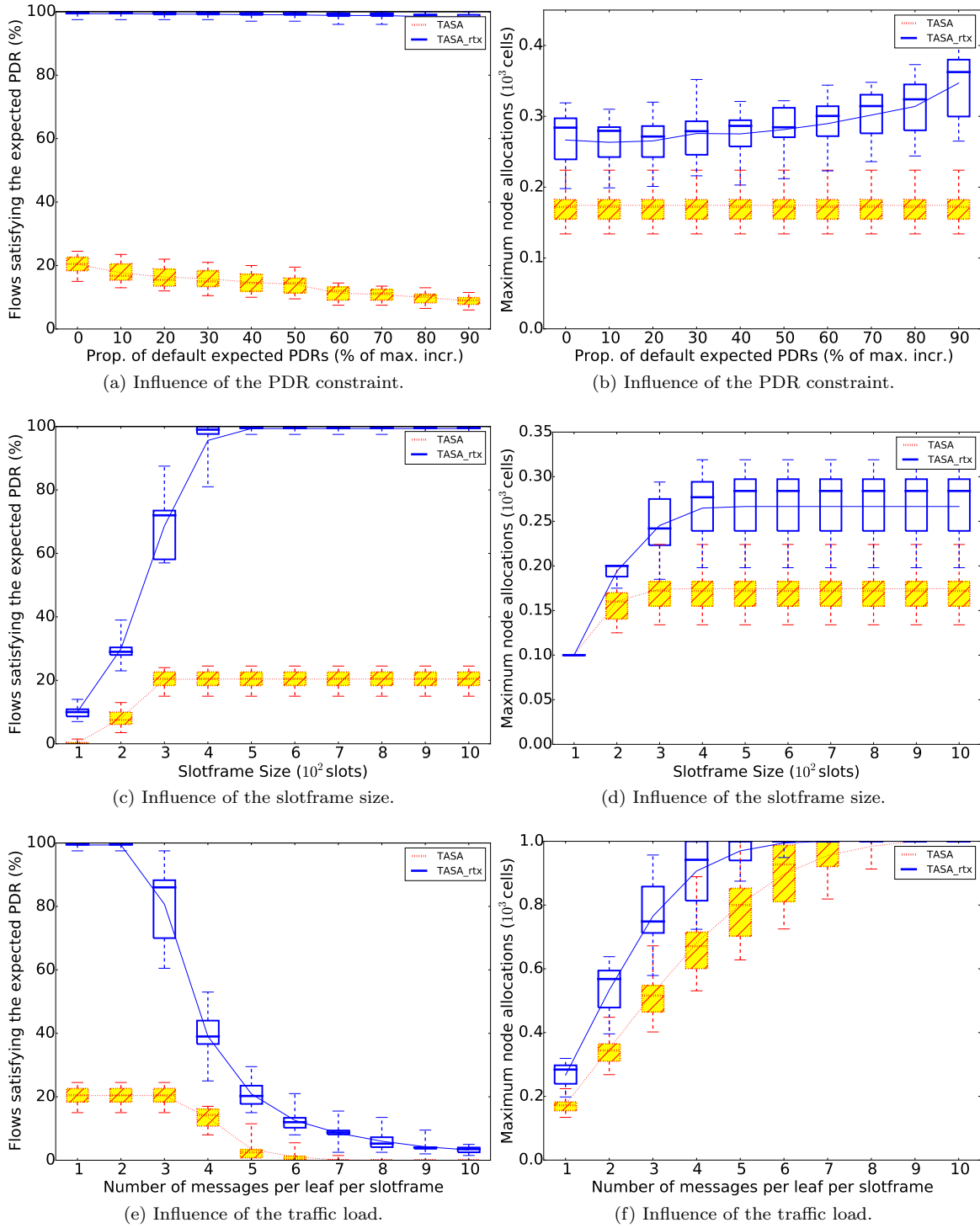


Figure 3: Evaluation of the performance in terms of PDR satisfaction and network resource usage.