From Best Effort to Deterministic Packet Delivery for Wireless Industrial IoT Networks

Remous-Aris Koutsiamanis, Georgios Z. Papadopoulos, *Member, IEEE*, Xenofon Fafoutis, *Member, IEEE*, Julián Martín Del Fiore, Pascal Thubert, and Nicolas Montavont

***Abstract*—Wireless industrial networks require reliable and deterministic communication. Determinism implies that there must be a guarantee that each data packet will be delivered within a bounded delay. Moreover, it must ensure that the potential congestion or interference will not impact the predictable properties of the network. In 2016, IEEE 802.15.4 time-slotted channel hopping (TSCH) emerged as an alternative medium access control to the industrial standards such as WirelessHART and ISA100.11a. However, TSCH is based on traditional collision detection and retransmission, and cannot guarantee reliable delivery within a given time. This paper proposes LeapFrog Collaboration (LFC) to provide deterministic and reliable communication over a routing protocol (RPL) based network. LFC is a novel multipath routing algorithm that takes advantage of route diversity by duplicating the data flow onto an alternate path. Simulations and analytical results demonstrate that LFC significantly outperforms the single-path retransmission-based approach of RPL + TSCH and the state-of-the-art LinkPeek solution.**

***Index Terms*—Deterministic networks, LeapFrog Collaboration (LFC), multipath routing algorithm, route diversity.**

I. Introduction

INDUSTRY 4.0 is an emerging domain of application for the Internet of Things (IoT), with goals to reduce the management cost and to contribute to the automation of the operational technology found in production chains in factories [1], [2]. Cost reduction can be achieved, in particular, by replacing the existing cables with a wireless medium, as long as an appropriate level of service for critical applications can still be guaranteed at all times. To that aim, the network must exhibit deterministic performance in terms of network reliability and timely delivery [2]–[4]. More precisely, an industrial communication framework must provide several nines of reliability in data delivery. For instance, several consecutive losses in an industrial automation control loop are sufficient to stop a production chain. Moreover, it should guarantee a worst case latency for a data packet across the network. This latency must be known in advance, and remain constant throughout the lifetime of the associated path. In order to replace wires, a wireless network should exhibit a high delivery ratio with an ultralow jitter, regardless of transient variations in the wireless medium and of the network congestion. However, the currently deployed IoT technologies focus on best-effort traffic, where the data packets may incur delays due to retransmission, queuing, and rerouting.

Time-slotted channel hopping (TSCH) is one of the medium access control (MAC) protocols defined in IEEE 802.15.4-2015 standard [5]. TSCH is a scheduled MAC-layer technology that is especially suited for industrial networks since it provides strict guarantees, i.e., low power, low delay, and reliable channel access. However, TSCH is prone to retransmissions when a data packet transmission fails, due to low link quality (i.e., multipath fading, distance) or external interferences, which may decrease reliability [6].

Scenarios such as factory automation, where the product can be for instance cars, require low power and lossy networks (LLNs) that consist of hundreds of sensors and actuators communicating LLN border router [7], [8]. In order to extend the network beyond the radio coverage of one node, a mesh technology enables a node to act as a relay for others, but, beyond one hop, it will require a protocol for routing packets throughout the network. The IPv6 routing protocol (RPL) for LLNs [9] is one of the most adopted RPL for the IoT. RPL builds a destination-oriented directed acyclic graph (DODAG) using a distance-vector technique whereby each node selects one or more parent(s), acting as a relay toward its root based on a common objective function (OF). The resulting acyclic topology is also used to route back to the node, using source routing in the so-called nonstoring mode (NSM), in which case the root is aware of the network topology. Phinney *et al.* [7] describe traffic patterns and network topologies in the industrial context and how RPL can provide the baseline protocol to address some specific applications. RPL is commonly used with 6TiSCH and is perceived as a solution for out-of-band industrial information transfer. As such, it is positioned for routing information, which

Manuscript received January 7, 2018; revised March 18, 2018 and June 4, 2018; accepted June 20, 2018. Date of publication July 18, 2018; date of current version October 3, 2018. Paper no. TII-18-0052. *(Corresponding author: Georgios Z. Papadopoulos.)*

R.-A. Koutsiamanis, G. Z. Papadopoulos and N. Montavont are with the IMT Atlantique, IRISA, UBL, France (e-mail: remous-aris.koutsiamanis@imt-atlantique.fr; georgios.papadopoulos@imt-atlantique.fr; nicolas.montavont@imt-atlantique.fr).

X. Fafoutis was with the University of Bristol, Bristol BS8 1TH, U.K. He is now with the Technical University of Denmark, Kongens Lyngby 2800, Denmark (e-mail: xefa@dtu.dk).

J. M. Del Fiore is with the ICube Laboratory, UFR de Mathematique et d'Informatique, Université de Strasbourg, Strasbourg 67400, France (e-mail: delfiore@unistra.fr).

P. Thubert is with the Cisco Systems, Mougins Cedex 06254, France (e-mail: pthubert@cisco.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2018.2856884

is not a part of the industrial process itself, but which is necessary auxiliary information for enhancing the industrial process. For example, diagnostics and asynchronous alerts are within the application field. We propose extensions to enhance reliability and determinism in such a context. The work presented in this paper is using RPL NSM as the baseline, and extends it with multipaths redundancy.

This paper investigates the forwarding mechanism and proposes to duplicate the data flow on alternate path, where multiple copies of the same data packet traverse on parallel paths through the wireless network. The proposed scheme, named LeapFrog Collaboration (LFC), allows combating potential link failures on a single path and exploiting path diversity in a wireless network to avoid retransmissions as much as possible. Since the first copy that arrives at the root is the one that matters, LFC lowers end-to-end delay performance. Furthermore, LFC comes with a topology-adapted scheduling algorithm that guarantees data delivery from the source to the destination within a slotframe (or not at all) and, thus, bounds jitter. Given the introduced overhead, we target lower bandwidth applications, such as critical monitoring or alerts. For these uses, LFC is implemented over IEEE 802.15.4-TSCH and RPL to reach network reliability above 99% while providing ultralow <15 ms jitter performance.

This paper extends [10], making the following additional contributions.

1) It comes with overprovisioning strategy, where one additional timeslot is reserved per data transmission, to guarantee the ultrahigh packet delivery ratio (PDR).
2) It introduces a scheduler that is adapted to the topology, whereby each data packet is delivered to the destination within one slotframe (101 timeslots in our examples).
3) It provides analytical expressions for the calculation of performance metrics of LFC, including the delay–jitter tradeoff, the PDR, and upper limits on the end-to-end delay and jitter.
4) It evaluates the robustness of the proposed scheme under various link qualities, on top of the COOJA simulator. In addition, LFC is compared to single-path retransmission-based approach (i.e., RPL + TSCH configuration) as well as the state-of-the-art solution LinkPeek [11].

II. Baseline Protocols and LFC Overview

A. IEEE 802.15.4-2015 TSCH

At its core, TSCH uses time-division multiple access and frequency hopping spread spectrum (FHSS) techniques to achieve high network reliability, reduce energy consumption, and mitigate multipath fading and the impact of external interference. In a TSCH network, nodes are constantly synchronized. The time is sliced into timeslots of equal length, sufficient enough to transmit a data packet and to receive an acknowledgment. A set of timeslots constructs a slotframe that repeats perpetually. Furthermore, an absolute sequence number (ASN) is assigned to each timeslot to count the number of timeslots since the establishment of the TSCH network. All nodes in the network are aware of the current ASN. To define a TSCH scheduler, for each radio link a collection of timeslots and channel offsets is

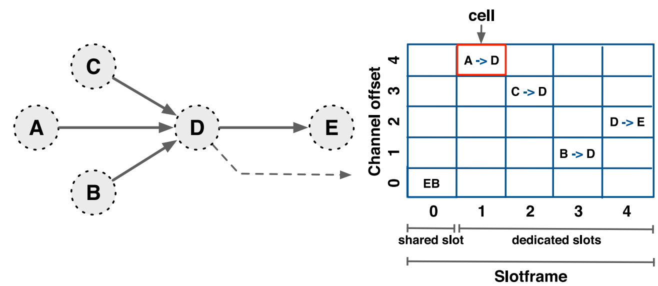

Fig. 1. Example of TSCH scheduling for node D. A → D stands for "*A transmits to D*," whereas EB cells are used for broadcast and advertisement frames.

assigned, called its cells. A channel offset is a "virtual channel" that is translated into a physical radio channel that is going to be employed for communication. The translation is carried out by an FHSS algorithm

$$\texttt{frequency} = F(\texttt{ASN} + \texttt{channelOffset})\,\%\,\texttt{nFreq} \quad (1)$$

where nFreq is the number of available physical channels (e.g., 16 when using IEEE 802.15.4-compliant radios at 2.4 GHz with all channels in use) [12]. F is a look-up table function that translates the result from the operation to actual radio channel (i.e., from 11th to 26th in 2.4-GHz band). In Fig. 1, a TSCH schedule is depicted.

B. Routing Protocol

RPL [9] is a distance-vector RPL specifically designed to manage hundreds of nodes in LLNs. At its core, RPL constructs a DODAG, i.e., a directed graph with no cycles. To this aim, RPL assigns to each device participating in the routing a *rank*, i.e., a metric that denotes the virtual distance from the root. Note that the rank is dynamically computed and constantly updated throughout the network lifetime. Different OF can be defined to compute a node's rank based on different types of metrics (hop count, link quality, expected retransmission times, etc.).

RPL initiates the DODAG construction from the root. It periodically broadcasts a DODAG information object (DIO), i.e., a control packet that includes certain configuration information as well as root's rank value. Nodes may discover their neighbors and their respective ranks upon the reception of such messages. Note that the frequency of DIO packets heavily depends on the network stability, i.e., the more stable is the network, the less frequently a DIO packet is transmitted. When a node receives a DIO message, it computes a new rank based on a given OF, and compares it with its current rank. Then, if the newly calculated rank is smaller than the current, it will add the transmitters address in its set of potential parents. From this set of parents, a node chooses its preferred parent as the node from which its rank will be minimal. If the new rank is bigger than the current rank, it is ignored. Once the network stabilizes, a node ends up with a preferred parent, a list of possible parents, children, and siblings. Then, a node may push its packets toward the root through its preferred parent. Note that once a node computed its own rank, based on the rank of its preferred parent and link quality, it periodically transmits its own DIO packets.

C. Motivation: Toward Deterministic Industrial Networks

A deterministic network ensures that data packets traverse the RPL network in a bounded window of time. It also guarantees that a periodic process will be repeated identically throughout the network lifetime. It particularly means that potential congestions or external interferences must not affect the predictable and deterministic behavior of the network.

As previously detailed, IEEE 802.15.4-2015 comes with resource reservation, i.e., transmissions are scheduled. However, wireless links are heavily affected by external interference and noise. Therefore, wireless communication comes with retransmission schemes, but at a cost of energy consumption, delay (in best-effort traffic) and bandwidth, since additional timeslots are required. In TSCH, if a data packet transmission fails, the transmitter will retransmit in the following slotframe, e.g., after 101 timeslots in our example, which is more than a second. In the case of a node crash, failures or over-the-air programming, the link quality between two nodes will drastically decrease (or eliminate the link) for some time, which will essentially increase the losses in the network. In such a scenario, retransmission-based schemes will not allow the packet to pass through this link. To overcome this issue, RPL comes with a failure solution wherein a child node will select another parent. However, the time needed for failure detection and new parent selection is large and during this time, all data packets will be discarded.

To address this limitation, this paper presents a novel technique, which takes advantage of path diversity and data duplication to combat the potential losses and to minimize the delay and jitter in wireless networks. It demonstrates that determinism can be ensured by using multiple parallel paths instead of retransmissions over the default DODAG path.

D. System Model and Assumptions

The context for LFC is an IEEE 802.15.4-2015 TSCH network running the RPL protocol for routing. The focus is on the transport of information from internal network nodes to the root node of the DODAG. The root node may be seen as a gateway to another network, possibly wired, and incoming packets can be sent to external destinations for further processing, but this is out of scope for this paper.

The information is in the form of user datagram protocol packet payloads. Therefore, at the transport layer the sending node does not expect or receive a delivery confirmation for the packets it sends, and thus no feedback loop exists at this layer. We are agnostic to the actual payload information transferred, which is considered an application layer concern.

The intended use is networks with at least two hops, since in one-hop networks simple retransmission provides the same performance. We consider only upward traffic (i.e., toward the DODAG root), which is typical of critical monitoring or alert information transmission. Given this use case, we mainly focus on reliability rather than throughput, so given the network overhead introduced for achieving high reliability, very high bandwidth applications are not an appropriate use case. In our analysis and simulations, we use symmetric wireless network links with a static (e.g., 80%) or variable (uniform 70%–100%) error probability, and in both cases packet loss follows a uniform distribution with the given probability. We rely on the default IEEE 802.15.4 TSCH retransmission scheme when packet losses occur; each hop acknowledges a data packet, and if the acknowledgment is not received, the sending node performs a retransmission. In our analysis and experiments, we use different maximum numbers of retransmissions. The network nodes themselves are assumed not to malfunction or operate maliciously and their internal clocks are assumed to be synchronized by IEEE 802.15.4-2015 TSCH (i.e., beacon packets and data packets with timing information to help correct drift as well as guard time in each timeslot to compensate for uncorrected drift [13]–[15]).

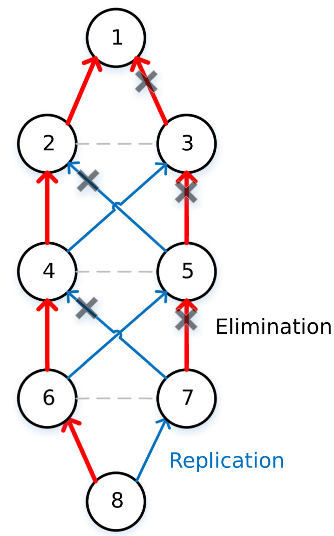

Fig. 2. LFC: Red arrows represent the RPL DODAG tree, whereas blue ones represent the alternative paths. Packet replication: 8 transmits twice the same packet, to its DP 6 and to its AP 7. Packet elimination: 5 discards the packet from 7, since it received it earlier from 6.

III. LFC Design

In this section, the LFC, a cross-layer (MAC and routing) scheme that minimizes the delay and the jitter metrics is detailed.

In a nutshell, LFC computes two parallel or interleaved paths for one track, with promiscuous listening between them, to allow the nodes on one path to overhear transmissions along the other path. Each node participating in a track selects a preferred or default parent (DP) and a so-called alternative parent (AP). For every data packet transmission within this track, the data packet is sent twice, one copy to the DP and another copy to the AP. For instance, in Fig. 2 a typical ladder-based topology of two parallel paths, $8 \rightarrow 6 \rightarrow 4 \rightarrow 2 \rightarrow 1$ and $8 \rightarrow 7 \rightarrow 5 \rightarrow 3 \rightarrow 1$, is depicted. In such a scenario, using packet replication and elimination (PRE) would allow transferring a copy of the packet along one or both of these paths, in a ship-by-night fashion.

The LFC algorithm can be designed and implemented in either a centralized or a distributed scheduling fashion. Under the centralized design, a central entity (e.g., the root) computes the routes and schedules the communication among the nodes in the network, similar to a label-switched path. Alternatively, in the distributed approach, each node constructs its path to the root, typically by employing a source routing header. In this paper, the latter case is investigated.

A. AP Selection

As previously mentioned, when running the RPL protocol, each node maintains a list of potential parents. LFC defines the preferred parent through the RPL DODAG as the DP, as shown in Fig. 2. To construct an alternative path toward the root, in addition to the DP, each node in the network registers an AP as well. There are multiple potential AP selection methods, but this paper presents a method that allows the two paths to remain correlated: a node will select an AP close to its DP in order to allow overhearing between parents. Thus, to choose an AP, a node will select another parent from its list of parents that has a common parent with the node DP. So the node will check if its default grandparent (DGP), the DP of its DP, is in the set of parents of a potential AP. If several potential APs follow this condition, the AP with the lowest rank will be selected.

B. LFC Operations

1) Packet Replication: To provide determinism in a wireless industrial network, LFC guarantees predictability in every level of the forwarding path. To that aim, under LFC, each node transmits (i.e., *replicates*) each data packet to its DP and its AP, respectively, in unicast transmission mode. Note that this procedure takes place at each level of the DODAG, between all relay nodes. In Fig. 2, the replication operation is illustrated, where node 6 is transmitting the data packet to both parents, nodes 4 (i.e., DP) and 5 (i.e., AP), in two different timeslots within the same TSCH slotframe. As a result, given the ladder-based topology, each data packet may traverse the wireless network over parallel paths. However, if multiple parents are not available, and thus no AP as well, LFC falls back to the normal RPL + TSCH operation, i.e., forwarding data to the preferred parent only, at the specific node where only one parent is available. Therefore, performance will be identical to the default RPL + TSCH case for that subset of the DODAG where only one parent is available.

2) Packet Elimination: By using the replication operation, it follows that a node may receive several copies of the same data packet, which increases the traffic load and, thus, may impact the network congestion and the impact on battery lifetime [16]. In order to avoid such a scenario, the *packet elimination* operation is introduced. Once a node receives the first copy of a data packet, it will discard the following received copies, as shown in Fig. 2. To do so, a sequence number is attached in each data packet to identify the duplicates. Note that the packet elimination operation is applied at each DODAG level.

C. LFC Features

1) Promiscuous Overhearing: LFC exploits the shared properties of the wireless medium to compensate for the potential loss that is incurred with radio communication. Considering that the wireless medium is broadcast by nature, then any neighboring node may listen to (i.e., overhear) a transmission if it is in the range of the sender, often called *promiscuous overhearing*. Thus, a given relay node may have more opportunities to receive a given data packet by listening to unicast transmissions toward other relaying nodes. This is a frequent occurrence in LFC since packet replication is also used. Furthermore, in the case where only one parent is available for a node, as with packet replication, the behavior will fall back to the normal TSCH+RPL case of performing no overhearing and thus being no worse than the default case. This fall back will be limited to the subset of the DODAG where only single parents are available.

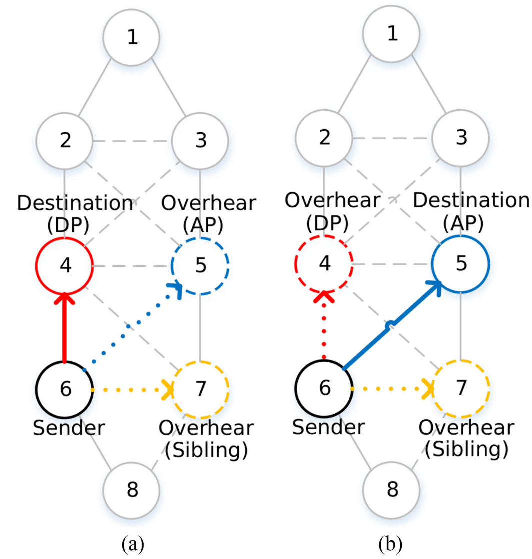

Fig. 3. By employing the overhearing operation, the DP, the AP, and the sibling nodes have more opportunities to receive the same data packet. (a) Unicast to DP. (b) Unicast to AP.

Since parents with a common ancestor are selected, a grandparent will have several opportunities to receive a given data packet by promiscuous overhearing. For instance, as shown in Fig. 3(a), when the intermediate node 6 is transmitting to its DP (i.e., node 4), the AP (i.e., node 5) may receive this data packet as well, and vice versa in Fig. 3(b). Thus, each parent (the DP and the AP) has twice the chances to receive a data packet for each transmission from each child: the original transmission directed to it and an overhearing from the child's transmission to its other parent. Finally, the probability of successful transmission from one DODAG level to its upper one can be enhanced, by considering the overhearing feature of the sibling nodes, i.e., the nodes that have the same parent as the transmitting node. For example, the transmission from node 6 to its DP 4, can be overheard not only by its AP, but also by its sibling as well, node 7. As a result, promiscuous overhearing not only improves network reliability, but may also decreases delay and jitter as long as the transmission opportunities from one level to another are grouped in the slotframe.

2) Overprovisioning: Using the previously described mechanisms (packet replication to an AP and promiscuous overhearing) together results in multiple opportunities for a packet to be

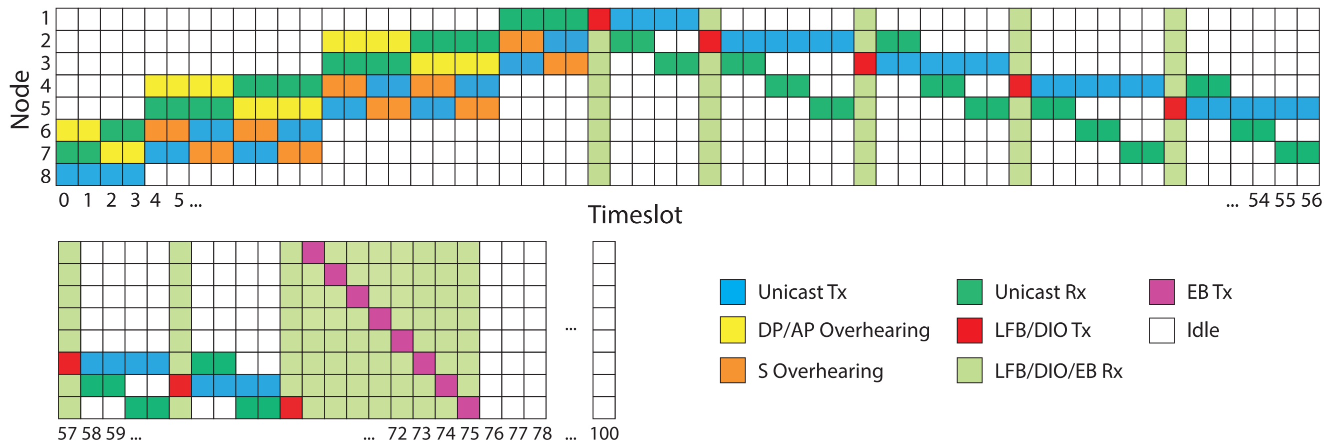

Fig. 4. Example of the adaptive LFC scheduler for eight nodes according to the topology in Fig. 3. It illustrates the following operations: replication, promiscuous overhearing, overprovisioning, as well as the enhanced beacons, LeapFrog beacon (LFB), and DIO control packets.

transmitted from one layer to the next. For instance, in Fig. 3, node 4 has up to four opportunities to receive the data packet: twice from node 6 (i.e., 1) through direct unicast transmission: 4 is the DP of 6, and 2) through overhearing the transmission from 6 to its AP node 5) and, similarly, it will receive twice from node 7. As a result, considering the ladder-based topology in Fig. 3, there are up to eight opportunities for a data packet to be received by the upper level of DODAG, i.e., four opportunities per node.

To further improve reliability, the concept of *overprovisioning* is introduced, which can be summarized as conditional retransmission. More specifically, each transmitter after sending a data packet will wait for an acknowledgment. In case of unsuccessful transmission, thanks to the overprovisioning algorithm, the transmitter will have another opportunity to transmit its data packet within the same slotframe, as shown in Fig. 4. It is worth mentioning that the overprovisioning timeslots are scheduled right after the original transmissions to bound the delay and, consequently, the jitter performance. For example, in timeslot 0, node 8 transmits unicast to node 7 while node 6 overhears, and in timeslot 1, these events are repeated in case a retransmission is needed. Thus, by introducing overprovisioning timeslots in the schedule, opportunities of successful packet reception are doubled.

3) Summary: Taking all the mechanisms used into account and used concurrently, the opportunities for reception belong to two groups: 1) if both the sending and the receiving levels have two nodes, then there are a total of 16 reception opportunities, otherwise, 2) for the levels that have either only one parent or only one sending node, there are a total of eight reception opportunities. These cases are summarized in Table I.

TABLE I
RECEPTION OPPORTUNITIES

Nodes / Level	Parents / Node	Overhearing	Over-provisioning	Total
2	2	2	2	$2^4 = 16$
1	2	2	2	$2^3 = 8$
2	1	2	2	$2^3 = 8$

IV. LFC SCHEDULER

A. Topology-Adapted Scheduler

Next, LFC comes with a scheduler adapted to the topology to achieve minimum and constant delay performance. To do so, the transmissions from the nodes far from the DODAG root are configured to be transmitted first. More precisely, the leaf nodes are configured to transmit first, and then relay nodes according to their distance from the root. As a result, all nodes access the wireless medium sequentially one after the other one and, therefore, the data packet may arrive at the root within one single slotframe, according to the requirements of deterministic networks in Industry 4.0 of ultralow jitter performance. The schedule is statically defined and precalculated offline for the given topology to be able to guarantee that the TSCH schedule will not externally affect the measurement of the metrics of the examined methods. It is also noteworthy that although in our description we use a schedule that contains all the cells that are required for the transmission of the data from the source to the root, nevertheless it is possible to use a smaller schedule if multislotframe delays can be tolerated by the application requirements. This is also a solution for networks that are so large so as to preclude the use of only one slotframe.

In Fig. 4, a scheduler with a single channel offset is illustrated, and it is adapted to eight nodes according to the topology in Fig. 3. As it can be observed, the leaf node 8 transmits first its data packet toward its DP, its RPL parent, while an additional timeslot is assigned for retransmission to the same destination in case of failed transmission. Then, another two timeslots are assigned to the leaf node to transmit to its AP (i.e., timeslots 2 and 3). Note that both parents are overhearing the others transmissions (see yellow boxes). Furthermore, it is worth mentioning that timeslots 4–11 are assigned to forward the data packet from level 3 to level 2 of DODAG tree, which is equal to 8 transmissions and 16 opportunities for a single data packet to reach the upper level of DODAG.

B. LFB Control Packet

During the network initialization phase, nodes receive all necessary RPL information through DIO messages. Thus, the nodes know their set of parents and, consequently, they decide their preferred parent. Similarly, LFC uses control packets called LFBs. These LFBs are periodically transmitted over the network in broadcast every 30 s. In the beginning, the LFB contains RPL-related information, i.e., a list of possible parents and the preferred parent. Based on this information, nodes eventually learn their DGP and, thus, they can define their AP, siblings, and other overheard neighborhood nodes. Later, this additional information is included in the LFB and consequently propagated. Hence, in case a node changes its DP, AP, or possible parents, the other nodes can learn about this change thanks to the LFBs and, consequently, recalculate their own AP. Finally, if a change was necessary, nodes will end up informing this to their neighbors with the upcoming LFBs. Thus, the update process travels across the network and the alternative paths can be kept dynamically available all the time.

V. Analysis

The first part of this section presents the delay–jitter performance tradeoff of TSCH. The goal is to provide insight to the reader on particular design decisions of LFC. The second part of this section provides analytical expressions for the calculation of several performance metrics of LFC, including the PDR, and upper limits on the end-to-end delay and jitter.

A. Delay–Jitter Tradeoff

In addition to promiscuous listening and exploiting multiple paths, LFC incorporates an extra timeslot for one retransmission in each link. Combining all these mechanisms, a particular packet has at least eight unique opportunities to progress to the next rank of the network within the same slotframe. Fundamentally, the number of unique opportunities a packet has to progress to the next rank is controlled by a performance tradeoff. Indeed, increasing this number is beneficial for the end-to-end jitter, as it increases the probabilities that a packet will reach the destination within a single frame. However, the more the allocated slots for retransmissions, the longer the frame is required to be. As a result, the allocation of additional timeslots for retransmissions within the same slotframe has a negative impact on the delay. This is the delay–jitter performance tradeoff.

Let us consider a TSCH neighborhood with one receiver (r) and N senders: $n_1, n_2, \ldots, n_N$. The number of consecutive timeslots per node per slotframe is denoted as k. For example, $k = 1$ corresponds to a frame size of N timeslots with one timeslot per node $(n_1 \rightarrow r, n_2 \rightarrow r, \ldots)$. Similarly, $k = 2$ corresponds to a frame size of $2N$ slots with two consecutive timeslots per node $(n_1 \rightarrow r, n_1 \rightarrow r, n_2 \rightarrow r, n_2 \rightarrow r, \ldots)$. Each transmission is modeled as an independent Bernoulli trial with the same probability of success, p (i.e., packet reception rate (PRR)), for each trial. For simplicity, let us assume infinite retransmissions. The probability that a packet will be successfully delivered after i failed attempts is given by

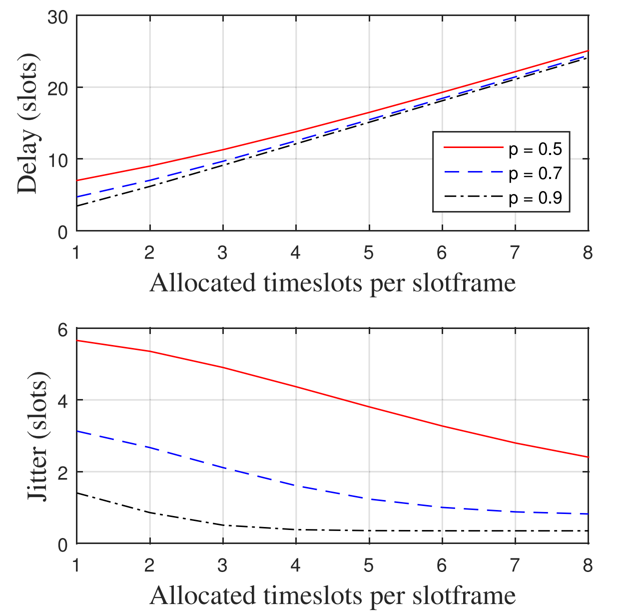

Fig. 5. Illustration of the delay–jitter tradeoff in a star TSCH neighborhood with four nodes. Increasing the allocated timeslots per slotframe decreases the jitter, yet increases the delay.

$$q_i = (1-p)^i p\,. \tag{2}$$

Without loss of generality, let us focus on the Nth node and let us assume that the packet is generated at the beginning of the frame. This assumption constitutes a worst case scenario for the delay. If the transmission occurs on the ith attempt, the delay is given by the following equation:

$$d_i = kN\left\lfloor \frac{i}{k} \right\rfloor + i \bmod k + k(N-1) \tag{3}$$

where $k(N-1)$ corresponds to the delay due to timeslots allocated to the other $N-1$ nodes in the first slotframe, $kN\left\lfloor \frac{i}{k} \right\rfloor$ corresponds to the delay of whole slotframes (when $i \geq k$), and $i \bmod k$ corresponds to the delay due to failed attempts in the last slotframe.

The average delay is calculated as the probability-based weighted sum of d_i, and it is given by

$$D = \sum_{i=0}^{\infty} q_i d_i \tag{4}$$

and the jitter, i.e., the deviation from perfect periodicity in packet delivery, be modeled as the standard deviation of the delay

$$J = \sqrt{\sum_{i=0}^{\infty} q_i (d_i - D)^2}\,. \tag{5}$$

Fig. 5 demonstrates the delay–jitter tradeoff for $N = 4$ and for various link qualities. It can be observed that as the number of timeslots per node per slotframe increases, the jitter decreases, yet with diminishing improvements. However, the delay increases rapidly in an almost linear fashion.

LFC balances the delay–jitter tradeoff by allocating up to $n^2 \times m$ timeslots for transmitting a given packet in the rank above, where $n = 2$ corresponds to the number of parents and $m = 2$ corresponds to the maximum number of transmission to a single parent.

B. LFC Performance in the Worst Case Scenario

This section models the performance of LFC in the worst case scenario, providing analytical expressions for calculating the end-to-end PDR, as well as the maximum end-to-end delay and jitter.

Let us consider that a packet is generated in the source node at the beginning of the frame in a TSCH network of R ranks. The number of parents (including any APs) is denoted by n. The maximum number of transmissions to a single parent within the same slotframe is denoted by m. From end to end, three cases can be identified. The first type corresponds to the source node i transmitting to its parents, which are located in rank $R-1$. The probability for the packet to fail to reach a parent j in rank $R-1$, denoted by q_{R-1}, is given by

$$q_{R-1}^{(j)} = p_{ij}^{mn} \tag{6}$$

where p_{ij} is the link-layer packet error probability for node i transmitting to parent j. It can be observed that (6) captures the fact that a packet has $n \times m$ opportunities to reach the next rank, as dictated by the LFC schedule, due to the feature of APs and retransmissions.

The probability for the packet to fail to progress to an intermediate rank $h \in [1, R-2]$ depends on if it managed to reach the previous rank

$$q_h^{(j)} = \prod_{i=1}^{n} \left(q_{h+1}^{(i)} + (1 - q_{h+1}^{(i)}) \cdot p_{ij}^{mn} \right) . \tag{7}$$

In the intermediate ranks, a packet has additional opportunities to reach the next rank due to the feature of promiscuous overhearing, captured in (7) by the product.

Finally, the third case corresponds to the last hop to the root. The probability for the packet to fail to reach the root, of rank 0, is given by

$$q_0 = \prod_{i=1}^{n} \left(q_1^{(i)} + (1 - q_1^{(i)}) \cdot p_{ij}^{m} \right) . \tag{8}$$

In this case, promiscuous overhearing is still possible, yet there are no APs.

Assuming that the packet is not retransmitted in a future frame, the end-to-end PDR is calculated recursively by $1 - q_0$. Note that (7) and (8) do not take into account the probability of sibling overhearing; thus, the model provides a lower bound of the end-to-end PDR.

Given the packet is successful at reaching the root within a slotframe, the end-to-end delay in the worst case scenario is provided by the following equation:

$$D_{\max} = 2 \cdot n \cdot m + (R-2) \cdot n^2 \cdot m \tag{9}$$

capturing the $n \times m$ timeslots reserved for the first (source to first intermediate rank) and final (last intermediate rank to destination) hops, as well as the $n \times n \times m$ timeslots reserved for the $R-2$ intermediate hops. Similarly, the end-to-end jitter in the worst case scenario is estimated by

$$J_{\max} = n \cdot m - 1 . \tag{10}$$

$D_{\max}$ and $J_{\max}$ are given in timeslots and they must be multiplied by the duration of the timeslot in order to be converted in time.

Fig. 6 plots $D_{\max}$ and $J_{\max}$ for networks of various depths (in terms of ranks, R) and widths (in terms of number of parents, n). The figure demonstrates how LFC scales with the network size. In particular, the delay increases linearly with the network depth and quadratically with the network width. The jitter is independent from the network depth and increases linearly with the network width.

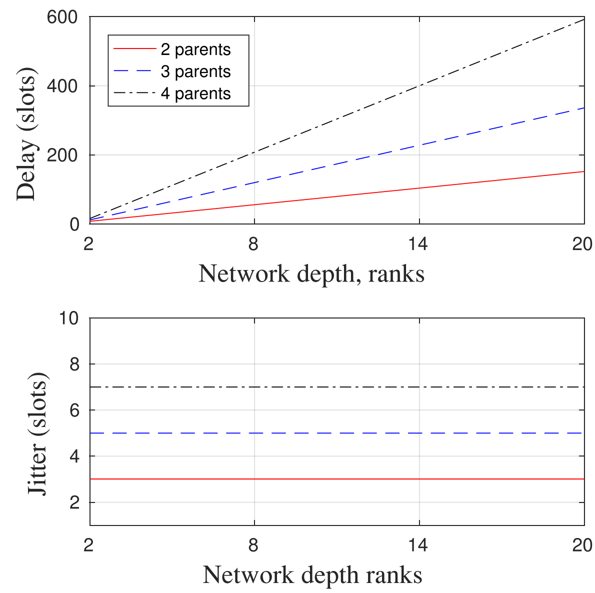

Fig. 6. Worst case scenario performance of LFC in networks of various depths (R) and widths in terms of parents (n). The delay increases linearly with the network depth and quadratically with the number of APs. The jitter is independent to the network depth and increases linearly with the number of parents.

C. Capacity Loss Tradeoff

The TSCH schedule constitutes a limiting factor for the maximum traffic that can traverse through the network. For example, if one timeslot per second is allocated to a particular link, the maximum supported traffic is one packet per second. This limiting factor is generally not considered an issue, as long as the overlaying application generates traffic at a low rate; yet, it can be a challenge when scheduling high-rate traffic [17].

More specifically, a TSCH schedule must overallocate timeslots to account for link-layer retransmissions [17]. For example, to support a link with a worst case $\mathtt{PRR} = 0.5$, the TSCH schedule should overallocate timeslots by a factor of expected transmission count $(\mathtt{ETX}) = 1/\mathtt{PRR} = 2$. In general, for a packet to traverse through a network of R ranks, a total number of $R \times \mathtt{ETX}$ timeslots are required. Note that, in practice, overallocation should be higher than the expected transmission count due to error busts and finite queues.

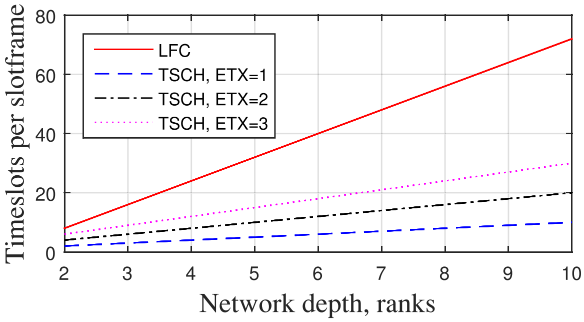

Fig. 7. Number of timeslots per slotframe required for a packet to traverse a network of R ranks ($n = 2$, $m = 2$). LFC sacrifices throughput for high reliability, low delay, and low jitter.

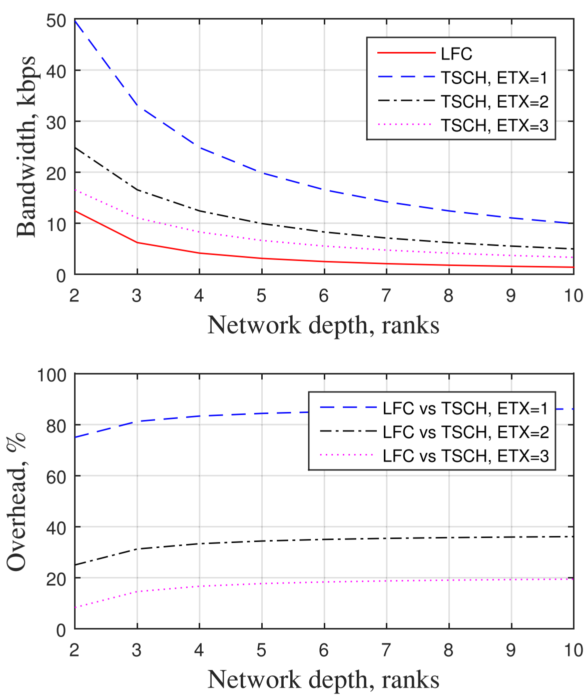

Fig. 8. Available end-to-end bandwidth for LFC and standard TSCH (top). The bandwidth overhead of TSCH compared to standard TSCH (bottom). LFC sacrifices throughput for high reliability, low delay, and low jitter.

LFC provides additional redundancy. Indeed, for a packet to traverse through R, the total number of required timeslots is given by the worst case end-to-end delay [see (9)]. Fig. 7 illustrates the number of timeslots required by LFC in contrast to standard TSCH. As demonstrated in the figure, LFC requires additional timeslots, imposing a limitation to the maximum achievable throughput. Therefore, LFC is suitable for industrial applications that can afford to trade bandwidth for high reliability, low delay, and low jitter.

This bandwidth overhead is better illustrated in Fig. 8. TSCH supports up to 100 timeslots of 10 ms per second; thus, up to 100 packets can be transmitted over a single TSCH link per second. This corresponds to an available bandwidth of

$$B_l = 100\ L \tag{11}$$

where L is the maximum packet size ($L = 127$ B or 1016 b in TSCH). Assuming for simplicity no downlink traffic, the available end-to-end bandwidth B is given by

$$B = \frac{B_l}{S} \tag{12}$$

where S is the timeslots per slotframe (see *y*-axis in Fig. 7). The available end-to-end bandwidth of LFC and standard TSCH for various network depths is plotted in Fig. 8 (top), whereas the overhead of LFC as compared to standard TSCH is shown in Fig. 8 (bottom). The figure shows that, under ideal conditions (no link-layer packet loss, $\texttt{ETX} = 1$), LFC introduces a considerable bandwidth overhead of more than 80%. In more realistic scenarios (i.e., $\texttt{ETX} = 2$ and $\texttt{ETX} = 3$), the bandwidth overhead of LFC is under 40% and 20%, respectively. Overall, LFC is able to support applications that require 2 kb/s end-to-end bandwidth in TSCH networks with depth of seven ranks or less.

VI. Performance Evaluation

A. Simulation Setup

To evaluate the performance of the LFC algorithm, COOJA, the network simulator distributed as part of the Contiki OS,[1] was employed.

The wireless network consists of eight nodes that are distributed in a typical ladder-based topology in an area of 130 m × 40 m, as depicted in Fig. 3. In particular, node 8 is the source, node 1 is the root, whereas nodes 2–7 are relays. The source node transmits one data packet every 15 s, using a 50 m transmission radius so that given the ladder topology, exactly a node's direct children, parents, and its sibling are reachable. The radios are configured to communicate at 250 kb/s. While this topology is relatively simple, it strikes a balance between complexity and simplicity, which has the advantages as follows:

1) it can be studied analytically as well (see Section V);
2) it has enough complexity and structure to highlight the advantages and disadvantages of our solution; and
3) it is very good predictor of performance in even more random topologies due to the method used for selecting APs in LFC (i.e., a node's DGP must be in the set of parents of a candidate AP).

At the routing layer, the RPL protocol [9] was used to build the DODAG, whereas at the MAC layer, IEEE 802.15.4-2015 TSCH was selected with slotframe length of 101 timeslots and one channel. The number of timeslots is generally configurable but we selected 101 since it is the default value and it is big enough for our cases. Any other big enough value would work as well. The duration of each timeslot is 10 ms, the lowest value 6TiSCH supports. This value provides the lower bound of the attainable delay for one network hop. The payload size is configured at 20 B. Each simulation lasts 41.25 h (not including the 25 min setup/initialization phase) within which approximately 9900 packets are sent, with five independent iterations per configuration. LFC was compared against that of the retransmission-based RPL + TSCH (i.e., a single copy

[1] http://www.contiki-os.org/start.html

TABLE II
SIMULATION SETUP

Topology	**Value**
Topology	Multi-hop, see Fig. 3
Number of nodes	8 (including the root)
Number of sources	1 source
Node spacing	40 m (in average)
Simulation	**Value**
Duration	41.25 $hours$
Traffic Pattern	1 $pkt/15\ sec$
Payload size	20 $bytes$
Routing model	RPL [8]
MAC model	TSCH [4]
TSCH	**Value**
EB period	16 sec
LB period	30 sec
Slotframe length	101
Timeslot length	10 ms

traverses the network), as well as the state-of-the-art solution LinkPeek [11]. Note that the maximum number of link-layer retransmissions was implemented under various configurations: 0, 2, 4, and 8, namely RT0, RT2, RT4, and RT8, respectively. For example, RT4 means that IEEE 802.15.4-TSCH configuration will retransmit at most four times if no acknowledgment was received at the link layer. In Table II, the details of the network parameters are presented.

It is noted that in this configuration (10 ms timeslot length, $R = 5$, $n = 2$, and $m = 2$), (9) and (10) indicate that the maximum delay is 240 ms and the maximum jitter is 30 ms, respectively.

B. Link Environment

In real-world deployments, the radio link quality presents dynamic behavior over time [18]. However, in critical industrial environments, the deployed wireless network has to combat such link quality variations to maintain ultrahigh network performance. In this study, the performance of the LFC was evaluated under various radio link qualities. In particular, the following four different use cases were studied: static link quality of 90%, 80%, 70%, and link quality uniformly random between 70% and 100%. In the last case, all link qualities are reset to a randomly generated value in the interval every 10 min. Additionally, the links between the sink node and its two children are always kept at 100%. The reason for this exception is because we would like to exclude these nodes from the reliability calculations. These nodes cannot support the LFC case due to having only one parent (the root node) instead of two. As a result, a lower link quality than 100% between these nodes and the root would unfairly affect LFC in comparison to the no-overhearing cases. Keeping the link qualities at 100% does affect latency, but in the same way for all the compared methods.

C. Simulation Results

In this section, the performance evaluation of LFC in terms of delay, jitter, reliability, and energy consumption is presented when compared against LinkPeek and default single-path RPL + TSCH with varying retransmissions at the link layer.

1) Delay: In Fig. 9(a), the mean and the standard deviation of MAC layer end-to-end, i.e., from the leaf node to the destination, delay is depicted. Note that the end-to-end delay includes the propagation time of the data packet, as well as the potential retransmission delay. As it can be observed, LFC achieves end-to-end delay close to 205 ms, which is in line with the proposed scheduler and with the mathematical analysis. Furthermore, considering Fig. 10, LFC demonstrates a very stable delay performance. When compared to the retransmission-based approaches of IEEE 802.15.4-TSCH (i.e., RT2, RT4, and RT8) and LinkPeek, LFC displays delay reduced by up to 577%, 669%, 664%, and 568%, respectively. Note that the nonretransmitting IEEE 802.15.4-TSCH RT0 approach does achieve very low delay ($\approx$204 ms) but at a very high cost in PDR. The delay is low because only delivered packets are considered, and if a packet has been delivered with RT0, by necessity, it will not have missed any transmissions along its path to delay it.

2) Jitter: To provide deterministic communication in a wireless network, it is necessary to obtain minimum jitter performance. Note that jitter is the variation in latency from one packet to another, here calculated as the standard deviation of end-to-end delay.

In Fig. 9(b), the average end-to-end jitter performance is illustrated. As it can be observed, LFC achieves ultralow jitter (i.e., $\approx$15 ms): lower than any other LFC or RPL + TSCH configuration and LinkPeek solution. On the other hand, the more retransmissions are configured for the following slotframe, the higher the jitter, i.e., see RPL + TSCH-RT4, RPL + TSCH-RT8. As a result, LFC decreases jitter by up to 7357%, 8799%, 8897%, and 7301% when compared to RPL + TSCH-RT2, RPL + TSCH-RT4, RPL + TSCH-RT8, and LinkPeek, respectively. Finally, note that the delay and jitter metrics were computed based only on successful packet receptions and also removing delays more than 3 standard deviations (3σ) away from the mean. This explains the performance of RPL + TSCH-RT0 ($\approx$14 ms) that presents high performance of jitter but extremely low PDR performance.

3) Packet Delivery Ratio: To further assess the performance of LFC, the network reliability was evaluated. To this aim, the PDR was computed, where packet loss is calculated as $1 - \text{PDR}$ and, thus, for example packet loss 0% is the equivalent of 100% PDR.

In Fig. 9(c), the PDR performance under various link qualities (i.e., link qualities fixed at 90%, 80%, $text$70% and variable random between 70% and 100%) is illustrated. LFC presents ultrahigh PDR performance above 99.1% in all cases and above 99.83% for cases with links with link quality above 80%. In fact, LFC demonstrates results similar to the theoretical performance, i.e., dark green column. These results can be explained as follows. As it can be observed, the additional features of LFC, overhearing and overprovisioning, fundamentally improve the performance of the LFC scheme, since by employing the overprovisioning option each data packet has an additional timeslot as a backup to retransmit in case of a failure of the original transmission (see Table I). As a result, LFC provides reliability similar to the theoretical and RPL + TSCH-RT8, while it minimizes

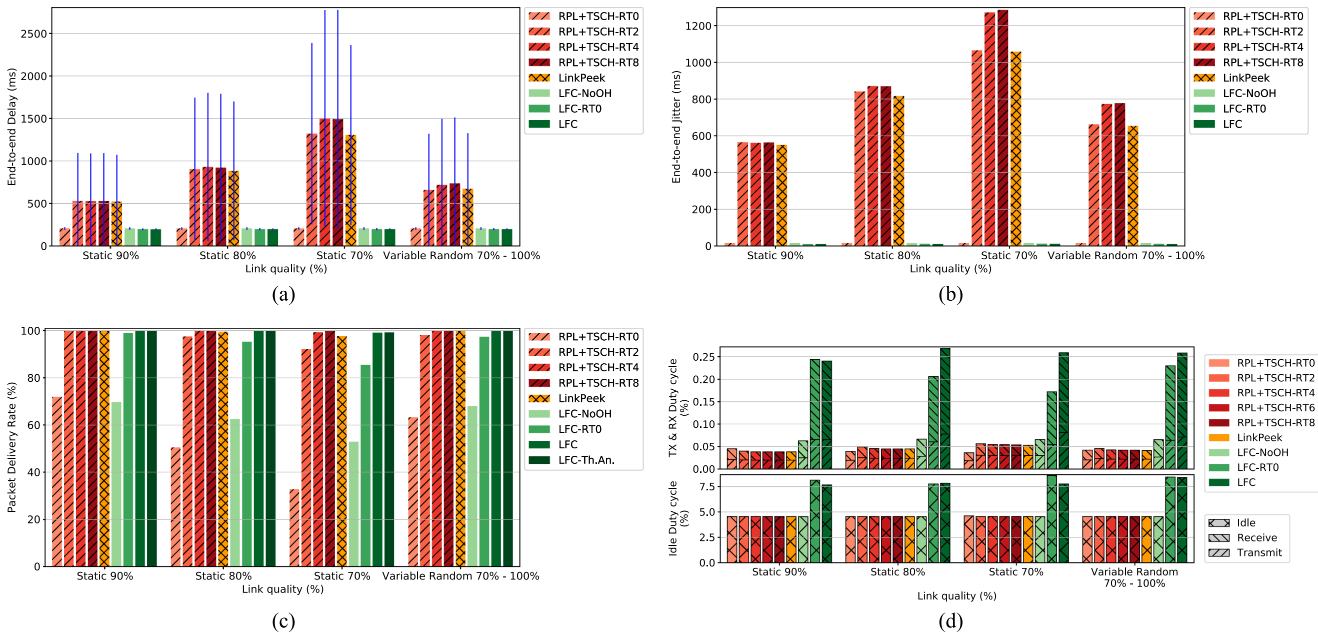

Fig. 9. Performance evaluation of LFC in terms of end-to-end delay, jitter, reliability, and radio duty cycle, when compared against single-path retransmission-based approaches of RPL + TSCH, state-of-the-art LinkPeek solution, and the theoretical PDR performance of LFC (LFC-Th.An.). (a) Mean and standard deviation of end-to-end delay. (b) Mean end-to-end jitter. (c) PDR performance. (d) Radio duty cycle performance for TX, RX, and Idle modes.

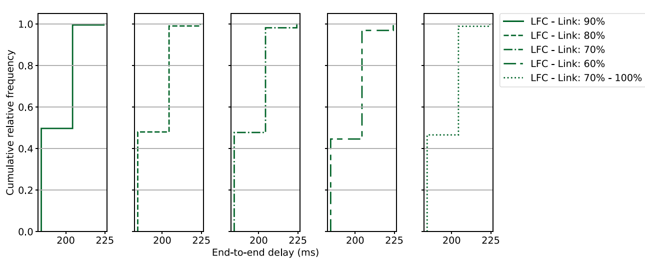

Fig. 10. Detailed representation LFC's delay performance under various link qualities.

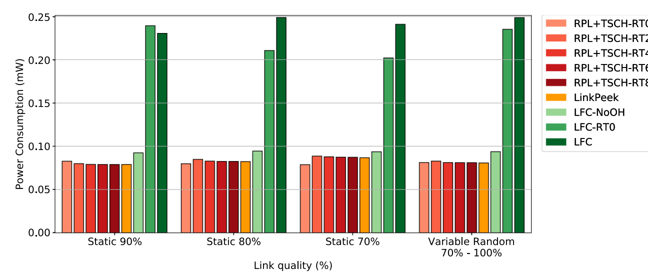

Fig. 11. Average radio power consumption on the Zolertia Z1 mote.

the jitter performance as well, contrary to the retransmission-based solutions.

4) Duty Cycle: Finally, this paper evaluates the energy consumption for all studied schemes.

Fig. 9(d) illustrates the average network-wide radio duty cycle (C) separately for the three modes (TX, RX, and Idle) that the radio can be in. The upper panel in Fig. 9(d) shows TX (C_{TX}) and RX (C_{RX}) stacked, whereas the lower panel shows only the Idle mode (C_{Idle}). For TX and RX, the maximum duty cycle is $\approx$0.27%, whereas for the Idle mode, the maximum is $\approx 8\%$. The majority of the duty cycle corresponds to the Idle mode, whereas TX and RX have very low and similar duty cycles.

The evaluation results utilize the fact that the majority of the energy consumption in IoT devices results from the operation of the radio module [19]. This leads providing duty cycle results, which are easily transferable to different hardware platforms, in lieu of power consumption values. As an example of this process, we provide the conversion from relative duty cycle to absolute power consumption for the Zolertia Z1 mote, which uses the CC2420 radio transceiver module. Given the values for radio power consumption for the Z1 [20] ($P_{\text{TX}} = 52.2\,\text{mW@3V}$, $P_{\text{RX}} = 56.4\,\text{mW@3V}$, $P_{\text{Idle}} = 1.28\,\text{mW@3V}$), we receive the average radio power consumption per mote as

$$E(P) = P_{\text{TX}}C_{\text{TX}} + P_{\text{RX}}C_{\text{RX}} + P_{\text{Idle}}C_{\text{Idle}}. \qquad (13)$$

The results for the Z1 mote using (13) are shown in Fig. 11 for each transmission method. The results indicate that LFC induces a straightforward impact on duty cycle performance, since LFC allocates additional timeslots per scheduler transmission to guarantee reliability above 99% and bounded delay performance (by using the overprovisioning feature). Moreover, by using the overhearing option, the nodes remain active for more timeslots to listen to neighboring transmissions, as compared to the default RPL + TSCH operation. Therefore, LFC consumes more network-wide energy when compared to LinkPeek and retransmission-based RPL + TSCH. As shown in Fig. 11, for the Z1 mote the overhead that LFC adds is $\approx$195% ($E(P_{\text{LFC}}) \approx 0.25\,\text{mW}$, $E(P_{\text{RPL+TSCH}}) \approx 0.085\,\text{mW}$). For other hardware platforms, the overhead introduced may differ since the reference energy consumption of the radio modes might be different from the Z1 mote, but our results facilitate calculating the overheads given the corresponding energy consumption values.

These results show that LFC provides determinism by bounding delay at the cost of energy consumption.

It is noted that one of the ongoing works on LFC is to reduce the unnecessary active timeslots by introducing a more intelligent scheduler. For instance, in the overprovisioning feature, it may not be necessary to explicitly allocate one additional timeslot per transmission, instead it could be configured probabilistically based on link quality, meaning that the more stable the link from A to B, the lower the probability to allocate an additional timeslot.

D. Feasibility for Industrial Applications

LFC has been implemented on Contiki OS with the constraints of low-power motes taken into consideration. The simulations for network and energy performance were executed using the COOJA target to allow examining a large set of parameters and to allow multiple iterations to increase statistical confidence in the results. However, to verify the feasibility of LFC for real industrial applications, we have used the Zolertia Z1 mote.

The Z1 mote has been selected because it is one of the most resource constrained motes available (16-b reduced instruction set computer CPU @16 MHz, 8 KB RAM, 92 KB internal flash memory, 2 MB external flash memory, CC2420 radio transceiver module), which can still execute in IPv6 mode to support 6TiSCH. It also has the added benefit that it is possible to simulate the Z1 mote using the MSPSim software package [21] within COOJA; thus, we have accurately verified that the operation is faithful to the scenarios examined in this paper. We are confident that since LFC using Contiki OS is operational on the very constrained Z1 mote, then it will be applicable to almost any other equivalent or more powerful hardware platform (e.g., CC2538 and CC2650).

More specifically, we have executed sample simulations to verify that the following statements hold.

1) LFC on Contiki OS compiles for the Z1 and fits within its available external flash memory (2 MB). More specifically, RPL + TSCH compiles to 495 960 B, whereas the LFC implementation to 499 600 B, so the overhead added is 3640 B or 0.73%.
2) LFC on Contiki OS compiles for and executes normally on the Z1, fitting within its available RAM (8 KB). The exact global RAM overhead is 99 B or 1.2% of RAM, augmented slightly by minimal stack use during function execution. Out of the 99 B, 40 B are arrays of configurable size and can be set to less to reduce memory usage. LFC does not use any dynamically allocated memory.
3) LFC on Contiki OS executes normally (no missed scheduled events or other timing issues) through the MSPSim simulator for the Z1. Thus, the computational overheads are acceptable. In practice, our implementation mainly adds functionality above the MAC layer, where timing is not so critical. It does modify the operation of the MAC packet reception handler, however our additions are executed *after* the packet has been received and not during the critical timeslot operation. The only modification inside the time critical timeslot operation handler is to enable and disable MAC address filtering when overhearing is required, which has no measurable impact on performance since it is just a configuration option change.

VII. Related Work

Phinney *et al.* [7] investigate the applicability of RPL for industrial applications and shows that RPL provides the baseline requirements, but QoS still needs to be addressed. Thus, it highlights the need for higher reliability and predictability. Attempting to provide wirelike reliability in industrial internet of things (IIoT) by utilizing the automatic repeat request method, commonly employed in general-purpose wireless networks, leads to simple and easily implemented solutions. However, while the value of simplicity and ease of implementation is considerable, the drawbacks of approaches of this nature are especially problematic for IIoT, which have strict constraints, i.e., increased delays and jitter metrics. Additionally, they decrease spectrum utilization by reserving part of it for the retransmissions.

In attempting to reduce the energy costs of reliability-enhancing methods, overhearing has been used to implicitly acknowledge the packet reception. In [22], Lee and Huh use this approach when the receiver in turn forward the packet to provide an acknowledgment of reception to the original sender. An enhanced version of this work by Maalel *et al.* [23] adds spatial diversity by maintaining an ordered list of neighboring nodes susceptible of retransmitting the packet.

Additionally, standardized methods of PRE targeting mission-critical and time-sensitive applications have been created by the Time-Sensitive Networking Task Group at the IEEE and the Deterministic Networking Working Group at the IETF. Both are working on redundancy-based methods wherein packet copies can follow noncongruent paths through the network while ensuring single delivery to the receiving end, even when one of the paths is interrupted [24].

Lohith *et al.* [11] propose an algorithm to increase reliability in IEEE 802.15.4 networks, by conditionally retransmitting packets to fallback RPL parents when link failure is detected. Their solution, however, does not aim at lowering end-to-end delay and by only retransmitting when failure is detected misses an opportunity for path diversification in cases where a node is completely disconnected from the network.

de Armas *et al.* [25] propose using data plane packet replication (up to the network graph degree) at the originating nodes only and routing the additional packet copies via disjoint paths, in order to improve reliability and latency, at the cost of increased energy consumption. They use a centralized scheduler to generate the TSCH schedule and a custom simulation for the performance evaluation, showing that packet replication significantly helps in achieving wirelike reliability.

Pavković *et al.* [26] propose the adaption of the IEEE 802.15.4 in order to facilitate its cooperation with the RPL protocol to achieve multipath routing. The adaptation is required due to a limitation in the IEEE 802.15.4, which allows only one associated parent at any time. Their work results in slightly improved end-to-end delay and reliability. However, their work does not take advantage of the multiple paths to replicate traffic in order to increase reliability.

Hosni and Théoleyre [27] propose a distributed scheduling algorithm for IEEE 802.15.4e-TSCH networks, which uses RPL ranks to group nodes into layers that can be scheduled in overlapping cells. Their aim is to use this to increase network capacity by taking spatial separation into account and to create schedules that feature an end-to-end upper bound to delay, specifically delivery within the same slotframe.

VIII. Conclusion

Industrial networks require deterministic protocols to guarantee that the transmitted data packet will traverse the wireless network in a predefined and constant delay. This paper proposes the LFC mechanism to exploit spatial diversity and packet redundancy to compensate for the inherently lossy wireless medium. At its core, LFC computes two parallel paths for a single data flow; thus, the nodes on one path may listen-in on the data transmissions along the other parallel path. As a result, each data packet gets multiple opportunities to be received at the upper DODAG level. The performance evaluation results demonstrate that the LFC achieves network reliability above 99%, while bounding the delay performance, i.e., providing an ultralow jitter performance of 15 ms. The ongoing work consists of further investigating a more sophisticated scheduler to reduce the unnecessarily active timeslots and, thus, to decrease the energy consumption. Furthermore, it is planned to investigate the performance of LFC in large-scale wireless networks. Finally, it is planned to investigate the behavior of LFC under realistic conditions by performing a set of experimental studies.

References

[1] E. Grossman *et al.*, "Deterministic networking use cases," Working Draft, IETF Secretariat, Internet-Draft draft-ietf-detnet-use-cases-13, Sep. 2017. [Online]. Available: https://tools.ietf.org/id/draft-ietf-detnet-use-cases-11.txt and https://tools.ietf.org/html/draft-ietf-detnet-use-cases-13

[2] L. D. Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Trans. Ind. Inform.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[3] S. Yamamoto, T. Emori, and K. Takai, "Field wireless solution based on ISA100.11a to innovate instrumentation," Yokogawa, Engl. ed., Tokyo, Japan, Tech. Rep. English Edition, vol. 53, no. 2, 2010.

[4] P. Thubert, "Converging over deterministic networks for an industrial internet," Ph.D. dissertation, Laurent Informatique, École nationale supérieure Mines-Télécom Atlantique Bretagne-Pays de la Loire, Brest, France, 2017. [Online]. Available: http://www.theses.fr/2017IMTA0011

[5] *IEEE Standard for Low-Rate Wireless Personal Area Networks*, IEEE Standard 802.15.4-2015 (Revision of IEEE Standard 802.15.4-2011), Apr. 2016.

[6] G. Z. Papadopoulos, A. Gallais, G. Schreiner, and T. Noel, "Importance of repeatable setups for reproducible experimental results in IoT," in *Proc. ACM Perform. Eval. Wireless Ad Hoc, Sensor, Ubiquitous Netw.*, 2016, pp. 51–59.

[7] T. Phinney, P. Thubert, and R. A. Assimiti, "RPL applicability in industrial networks," Working Draft, IETF Secretariat, Internet-Draft draft-ietf-roll-rpl-industrial-applicability-02, Oct. 2013.

[8] K. Pister, P. Thubert, S. Dwars, and T. Phinney, "Industrial routing requirements in low-power and lossy networks," IETF, RFC 5673, 2009.

[9] T. Winter *et al.*, "RPL: IPv6 routing protocol for low-power and lossy networks," IETF, RFC 6550, 2012.

[10] G. Z. Papadopoulos, T. Matsui, P. Thubert, G. Texier, T. Watteyne, and N. Montavont, "Leapfrog Collaboration: Toward determinism and predictability in industrial-IoT applications," in *Proc. IEEE Int. Conf. Commun.*, 2017, pp. 1–6.

[11] Y. S. Lohith, T. S. Narasimman, S. V. R. Anand, and M. Hedge, "Link peek: A link outage resilient IP packet forwarding mechanism for 6LoWPAN/RPL based low-power and lossy networks (LLNs)," in *Proc. IEEE Int. Conf. Mobile Serv.*, Jun. 2015, pp. 65–72.

[12] T. Watteyne, M. Palattella, and L. Grieco, "Using IEEE 802.15.4e time-slotted channel hopping (TSCH) in the internet of things (IoT): Problem statement," RFC 7554, 2015.

[13] D. Stanislowski, X. Vilajosana, Q. Wang, T. Watteyne, and K. S. J. Pister, "Adaptive synchronization in IEEE802.15.4e networks," *IEEE Trans. Ind. Inform.*, vol. 10, no. 1, pp. 795–802, Feb. 2014.

[14] T. Chang, T. Watteyne, K. Pister, and Q. Wang, "Adaptive synchronization in multi-hop TSCH networks," *Comput. Netw.*, vol. 76, pp. 165–176, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128614003922

[15] G. Z. Papadopoulos *et al.*, "Guard time optimisation and adaptation for energy efficient multi-hop TSCH networks," in *Proc. IEEE 3rd World Forum Internet Things*, 2016, pp. 301–306.

[16] G. Z. Papadopoulos, J. Beaudaux, A. Gallais, P. Chatzimisios, and T. Noel, "Toward a packet duplication control for opportunistic routing in WSNs," in *Proc. IEEE Global Telecommun. Conf.*, 2014, pp. 94–99.

[17] A. Elsts, X. Fafoutis, J. Pope, G. Oikonomou, R. Piechocki, and I. Craddock, "Scheduling high-rate unpredictable traffic in IEEE 802.15.4 TSCH networks," in *Proc. 13th Int. Conf. Distrib. Comput. Sensor Syst.*, 2017, pp. 3–10.

[18] G. Z. Papadopoulos, A. Gallais, G. Schreiner, E. Jou, and T. Noel, "Thorough IoT testbed characterization: From proof-of-concept to repeatable experimentations," *Comput. Netw.*, vol. 119, pp. 86–101, 2017.

[19] J. Eriksson, F. Österlind, N. Finne, A. Dunkels, N. Tsiftes, and T. Voigt, "Accurate network-scale power profiling for sensor network simulators," in *Proc. 6th Eur. Conf. Wireless Sensor Netw.*, 2009, pp. 312–326.

[20] Zolertia, Z1 Datasheet, Mar. 2010.

[21] J. Eriksson, A. Dunkels, N. Finne, F. Österlind, and T. Voigt, "MSPSim—An extensible simulator for MSP430-equipped sensor boards," in *Proc. Eur. Conf. Wireless Sensor Netw.*, 2007, vol. 118, pp. 41–42.

[22] G.-W. Lee and E.-N. Huh, "Reliable data transfer using overhearing for implicit ACK," in *Proc. ICCAS-SICE*, 2009, pp. 1976–1979.

[23] N. Maalel, P. Roux, M. Kellil, and A. Bouabdallah, "Adaptive reliable routing protocol for wireless sensor networks," in *Proc. Int. Conf. Wireless Mobile Commun.*, 2013, pp. 52–55.

[24] G. Z. Papadopoulos, N. Montavont, and P. Thubert, "Exploiting packet replication and elimination in complex tracks in 6TiSCH LLNs," Working Draft, IETF Secretariat, Internet-Draft draft-papadopoulos-6tisch-pre-reqs-00, Jul. 2017.

[25] J. de Armas, P. Tuset, T. Chang, F. Adelantado, T. Watteyne, and X. Vilajosana, "Determinism through path diversity: Why packet replication makes sense," in *Proc. Int. Conf. Intell. Netw. Collaborative Syst.*, 2016, pp. 150–154.

[26] B. Pavković, F. Theoleyre, and A. Duda, "Multipath opportunistic RPL routing over IEEE 802.15.4," in *Proc. ACM Int. Conf. Model., Anal. Simul. Wireless Mobile Syst.*, 2011, pp. 179–186.

[27] I. Hosni and F. Théoleyre, "Self-healing distributed scheduling for end-to-end delay optimization in multihop wireless networks with 6TiSCh," *Comput. Commun.*, vol. 110, pp. 103–119, 2017.

Remous-Aris Koutsiamanis received the B.Sc. degree (with distinction) in informatics from the University of Piraeus, Piraeus, Greece, in 2005, the M.Sc. degree (with a full scholarship) in artificial intelligence from the University of Edinburgh, Edinburgh, U.K., in 2006, and the Ph.D. degree (with Hons.) in distributed management of competitive access to common resources using algorithmic game theory from the Department of Electrical and Computer Engineering, Democritus University of Thrace, Komotini, Greece, in 2016.

He is currently a Postdoctoral Researcher with the IMT Atlantique Bretagne-Pays de la Loire, Rennes, France, and a Visiting Fellow with Bournemouth University, Bournemouth, U.K., where he was previously a Postdoctoral Research Assistant. He has participated in numerous international and national research projects. He has been involved in the organization of an international event (AdHoc-Now'18) and has served as a Reviewer in multiple conferences and journals. His research interests include the Industrial IoT, 6TiSCH, and network resource allocation within the wider networking field.

Georgios Z. Papadopoulos (S'10–M'15) received the B.Sc. degree in informatics from Alexander Technological Educational Institute of Thessaloniki, Thessaloniki, Greece, in 2011, the M.Sc. degree in telematics engineering from the Carlos III University of Madrid, Madrid, Spain, in 2012, and the Ph.D. degree (with honors) in improving medium access for dynamic wireless sensor networks from the University of Strasbourg, Strasbourg, France, in 2015.

He is currently an Associate Professor with the IMT Atlantique Bretagne-Pays de la Loire, Rennes, France. He was a Postdoctoral Researcher with the University of Bristol. His research interests include Industrial IoT, 6TiSCH, LPWAN, battery management system, and smart grid.

Dr. Papadopoulos has participated in various international and national (FP7 RERUM, FIT Equipex) research projects. He has received the prestigious French national ANR JCJC grant for young researchers. He has been involved in the organization of many international events (AdHoc-Now'18, IEEE CSCN'18, IEEE ISCC'17). Moreover, he has been serving as an Editor for *Wireless Networks* journal and Internet Technology Letters. He was the recipient of the Best Ph.D. Thesis Award granted by the University of Strasbourg and two Best Paper Awards (IFIP Med-Hoc-Net'14 and IEEE SENSORS'14).

Xenofon Fafoutis (S'09–M'14) received the B.Sc. degree in informatics and telecommunications from the University of Athens, Athens, Greece, in 2007, the M.Sc. degree in computer science from the University of Crete, Heraklion, Greece, in 2010, and the Ph.D. degree in embedded systems engineering from the Technical University of Denmark, Kongens Lyngby, Denmark, in 2014.

From 2014 to 2018, he held various researcher positions with the University of Bristol, Bristol, U.K., and he was a core member of SPHERE: UK's flagship Interdisciplinary Research Collaboration on Healthcare Technologies. Since 2018, he has been an Assistant Professor with the Embedded Systems Engineering Section, Department of Applied Mathematics and Computer Science, Technical University of Denmark. His research interests include networked embedded systems as an enabling technology for digital health, smart cities, Industry 4.0, and the Internet of Things.

Julián Martín Del Fiore received the Electronics Engineering diploma (with Hons.) in deterministic networks for IIoT applications from the University of Buenos Aires, Buenos Aires, Argentina, in 2017. He is currently working toward the Ph.D. degree (first year) in verifiable data plabe at the ICube Laboratory, University of Strasbourg, Strasbourg, France.

His research focuses on proposing solutions to cope with routing insecurity on the Internet. He also did an internship in the area of Industrial IoT centered on 6TiSCH networks at IMT Atlantique. His research interests include Internet and machine learning.

Pascal Thubert received the Engineering degree from École Centrale de Lyon, Lyon, France, in 1987, and the Ph.D. degree in converging over deterministic networks for an industrial internet from IMT Atlantique Bretagne-Pays de la Loire, Rennes, France, in 2017.

Since joining Cisco, Mougins Cedex, France, in 2000, he has been actively involved in research, development, and standards efforts on Internet mobility and wireless technologies. He currently works at Cisco's Chief Technology and Architecture office, where he focuses on products and standards in the general context of IPv6, wireless, and the Internet of Things. He Co-Chairs 6TiSCH, the IETF Working Group focusing on IPv6 over the 802.15.4 TSCH deterministic MAC, and LPWAN, that applies IETF protocols over low power wide area networking technologies. Earlier, he specialized in IPv6 as applied to mobility and wireless devices and developed routers and switches microcode in Cisco's core IPv6 product development group. In parallel with his R&D missions, he has authored multiple IETF RFCs and draft standards dealing with IPv6, mobility and the Internet of Things, including NEMO, 6LoWPAN, and RPL.

Nicolas Montavont received the M.Sc. and Ph.D. degrees in computer science from the University of Strasbourg, Strasbourg, France, in 2001 and 2004, respectively, and the HDR degree in mobility support in networks from Université de Rennes 1, Rennes, France, in 2015.

He is currently a Full Professor with the SRCD Department, IMT Atlantique Bretagne-Pays de la Loire, Rennes, France, and has been responsible for the IRISA OCIF team since 2015. He also did a Postdoc with the National Institute of Standard and Technologies, Gaithersburg, MD, USA. His research interests include mobility and multihoming management in IPv6 networks, wireless communications, and Internet of Things.