Autonomous and Traffic-aware Scheduling for TSCH Networks

Sana Rekik, Nouha Baccour, Mohamed Jmaiel, Khalil Drira, Luigi Alfredo Grieco

Autonomous and Traffic-aware Scheduling for TSCH Networks

Sana Rekik, Nouha Baccour, Mohamed Jmaiel
ReDCAD Laboratory
University of Sfax, Tunisia
{sana.rekik, nouha.baccour, mohamed.jmaiel}@redcad.org

Khalil Drira
LAAS-CNRS
Univ. de Toulouse, France
khalil.drira@laas.fr

Luigi Alfredo Grieco
DEI
Politecnico di Bari, Italy
a.grieco@poliba.it

***Abstract*—Wireless Sensor Networks (WSNs) have been recognized as a promising communication technology for smart grid monitoring and control applications. However, the deployment of WSNs in smart grid brought new challenges that pertain to the harsh electrical grid nature, and the different and often contradicting communication requirements of smart grid monitoring applications. MAC protocols play a crucial role to meet the reliability and latency requirements of WSN-based smart grid communications. In particular, the IEEE 802.15.4 TSCH (Time Slotted Channel Hopping), the latest generation of low-power and highly reliable MAC protocols, orchestrates the medium access according to a time-frequency communication schedule. However, TSCH specification does not provide any practical solution for the establishment of the schedule. Orchestra is a recent scheduling solution for TSCH that brings significant advantages such as, the use of simple scheduling rules, the low signaling overhead, and the high delivery ratio. Despite its unique features, Orchestra has the limitation of computing the TSCH schedule at each node independently from its traffic load, which can drastically affect the communication delay. This limitation makes Orchestra not sufficiently convenient for several delay-sensitive smart grid applications. Further, the current TSCH specification does not support traffic differentiation (i.e. handle all packets equally regardless of their criticality levels). In this paper, we propose an enhanced Orchestra-based TSCH protocol, called e-TSCH-Orch, that dynamically adjusts time slots assignment according to traffic load and criticality level. The performance analysis of e-TSCH-Orch shows that it significantly reduces the communication delay compared to the original Orchestra-based TSCH, while preserving the low signaling overhead and the high packet delivery ratio.**

Keywords— Wireless sensor network, smart grid, TSCH MAC, Orchestra scheduling, RPL routing, traffic differentiation, communication delay.

I. Introduction

WSNs have been recognized as a promising communication technology for the Internet of Things (IoT). In particular, smart grid applications rely on WSNs for enabling the pervasive monitoring and control of electric grid networks from the power generation plants to the transmission and distribution systems [1, 2]. The deployment of WSNs in smart grids brings new challenges. For instance, electric grid environments are typically characterized by highly corrosive conditions (e.g. rain, humidity, electric equipment's noise, electromagnetic interference, obstructions and vibrations), which turns radio links extremely unreliable and contributes to sensor nodes failures [3]. Further, WSN-based smart grid applications have different and often contradictory QoS requirements in terms of reliability and latency [4]. For example, overhead transmission line monitoring [5] and substation automation [5] represent critical smart grid applications that require an almost deterministic (and highly reliable) service, whereas other applications, such as Demand Response and Advanced Metering Infrastructure (AMI) [6], entails more relaxed requirements. Importantly, the QoS requirements can also differ within the same application [7], according to the importance of the sensed information (e.g. temperature information, vibration information and alarms). The resulting heterogeneous data traffic can be classified into different classes, where each traffic class has particular QoS requirements (e.g. in terms of latency and reliability) and also particular characteristics (e.g. data rate and traffic distribution).

The challenges raised by the application of WSNs in smart grids should be properly considered in the design of efficient communication protocols. Especially, MAC protocols play a crucial role to meet the reliability and latency requirements of WSN-based smart grid applications. The Time Slotted Channel Hopping (TSCH), part of the IEEE 802.15.4 standard [8], represents the latest generation of low-power and highly reliable MAC protocols. It has been specifically proposed to meet the requirements of industrial applications, including smart grid applications. Further, the IEEE 802.15.4 TSCH along with the IETF 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks) [9] and RPL (Routing Protocol for Low Power and Lossy Networks) [10] protocols are considered as key building blocks of the emerging 6tisch architecture [11], under standardization within the IETF 6tisch Working Group. These features motivate us to investigate the use of TSCH protocol for WSN-based smart grid applications.

With TSCH, channel hopping is added on top of time slotted MAC in order to counteract frequency selective fading and improve the reliability of radio links. TSCH orchestrates the medium access according to a communication schedule that indicates to each node what to do in each slot and frequency channel: transmit, receive, or sleep. The IEEE 802.15.4 standard only specifies how the MAC layer executes the schedule. However, it does not define how to establish it. Therefore, several scheduling algorithms have been proposed for TSCH networks. They can be broadly classified as either centralized [12–14] or distributed [15–

18] algorithms. Both centralized and distributed scheduling techniques involve additional communication overhead: each node is supposed to communicate with its neighbors or with a central entity to exchange network and traffic information in addition to scheduling information. Orchestra [19] is a recent scheduling technique that is neither centralized nor distributed. Rather, it is referred to autonomous scheduling technique as each node autonomously builds its own schedule without any negotiation with its neighbors. The schedule is computed based on available routing information and is automatically updated whenever the routing topology evolves. Orchestra presents several strengths making it a promising scheduling solution for TSCH networks. It is able to build schedules, using simple scheduling rules, without triggering a signaling overhead. Further, it was proven that Orchestra can achieve high delivery ratio [19]. These features motivate us to use TSCH MAC protocol with Orchestra scheduling approach for WSN-based smart grid applications.

In this paper, we first analyze the performance of TSCH MAC using Orchestra for schedule establishment and we show that TSCH may experience high end-to-end delay, particularly when the traffic is not uniformly distributed in the network. As a matter of fact, the amount of data traffic to deliver to the root depends on the sensor node location in the routing tree-like topology. Generally, the more the node is close to the root the more data it has to forward. However, Orchestra allocates the same number of time slots to all nodes, which leads to additional communication delays due to packets enqueuing in the buffers of congested nodes. This limitation makes Orchestra not sufficiently convenient especially for smart grid communication where low latency is a key requirement for several applications [6]. On the other hand, the current TSCH specification does not support traffic differentiation, i.e. handle all packets equally regardless of their criticality levels. However, WSN-based smart grid applications have different and often conflicting requirements in terms reliability and latency, resulting in heterogeneous data traffic patterns.

To overcome the above limitations, we propose to optimize TSCH protocol using Orchestra and design a more efficient MAC that dynamically adjusts time slots assignment according to traffic demand and criticality levels, while preserving the strengths of the original Orchestra in terms of low communication overhead and performance. The resulting enhanced protocol, that we call e-TSCH-Orch, is validated through both simulations and real experiments. Analysis results show that e-TSCH-Orch significantly reduces the end-to-end communication delay compared to the original protocol, while preserving (or improving in particular scenarios) the packet delivery ratio. It also favors the delivery of high priority traffic.

The rest of this paper is organized as follows. In the next section, we provide necessary background information on TSCH MAC, Orchestra scheduling, and RPL routing. A discussion of recent works related to TSCH scheduling algorithms is then given in Section III. In Section IV, we show evidence of the communication delay caused by Orchestra. In Section V, we describe the enhanced Orchestra-based TSCH protocol. In Section VI, we assess the effectiveness of the proposed solution through simulation and experiment results. Finally, we conclude in Section VII.

II. Background

In this section, we give a short overview of TSCH MAC and RPL routing protocols followed by a description of Orchestra, a promising scheduling solution for TSCH.

A. TSCH protocol

The TSCH protocol combines three mechanisms, namely time slotted access, multi-channel communication and channel hopping, thus mitigating the unpredictable behavior of radio links in industrial environments. Time slotted mechanism consists in dividing time into slots that are grouped into a repetitive slotframe structure used to synchronize nodes. The slotframe indicates to each node what to do in each slot: transmit, receive, or sleep. For each transmission/reception slot, the schedule tells the node which neighbor to communicate with and on which channel offset. Thus, slotted access allows the minimization of packet collisions and enables the support of almost deterministic services. Using multi-channel communication, several nodes can communicate at the same time using different channels, which increases the network capacity. The channel hopping mechanism allows nodes to change the communication channel among a predefined sequence of channels. The same slot translates into a different frequency at each slotframe instance, thus minimizing the effects of interference and improving reliability.

B. RPL protocol

The RPL routing protocol [10] is designed for Low power and Lossy Networks (LLNs) such as WSNs. It organizes the network in form of a Directed Acyclic Graph (DAG), rooted at a small set of LLN sinks. For each sink, a Destination Oriented DAG (DODAG) is created. The establishment and the maintenance of the DODAG are ensured by DIO (DODAG Information Object) control messages disseminated using a trickle timer. The DIO packets contain information such as the metrics used for path cost computation and the Objective Function, which maps the optimization requirements of the target scenario. In this paper, we consider the simplest topology with a single sink, also referred as *DODAGroot*. Although RPL can manage several kinds of traffic flows (to and from the *DODAGroot* or between any pair of nodes in the network), we have focused on the dominant multipoint-to-point traffic, i.e. flowing from the nodes in the network towards the *DODAGroot*, which is more related to monitoring applications in industrial environments.

RPL employs a gradient strategy, which introduces the concept of *rank* to define the individual position of a node with respect to its *DODAGroot*. A fundamental property of an RPL-organized network is that the *rank* should monotonically decrease along the DODAG and towards the destination, in accordance to the gradient-based approach. In general, the

rank is computed based on path metrics, but it is used to let the routing topology being loop-free.

RPL can adopt several metrics for computing the *rank*. The default RPL metric is ETX (Expected Transmission Count). In this paper, the Opt-FLQE$_{RM}$ [20] metric is used as it was shown to improve significantly the RPL reliability compared to the default RPL metric, ETX.

C. Orchestra schedule

Orchestra is a scheduling approach for TSCH and RPL networks. The Orchestra schedule is computed in an autonomous fashion, where each node in the network locally maintains its own schedule based on information from the routing layer. Specifically, each node's schedule consists of different slotframes having different lengths (periods). Each slotframe is assigned to a particular traffic plane: TSCH MAC, RPL routing, and application. The MAC slotframe schedules the transmissions of TSCH beacons that ensure TSCH association and parent-child synchronization. The routing slotframe schedules the transmissions of broadcast RPL signaling, such as the DIO messages, while the application slotframe is used for unicast data traffic transmissions from any node to its RPL parent. The slotframe lengths are mutually prime. Each slotframe has a unique identifier (denoted handle), such as the smaller the handle, the higher the priority of the slotframe. If slots from different slotframes overlap, the slot of the highest priority slotframe takes precedence. Figure 1 illustrates an example of Orchestra schedule containing three slotframes – MAC, routing and application slotframes– of lengths 5, 3, and 2 slots respectively. In this example, priority increases from top to bottom.

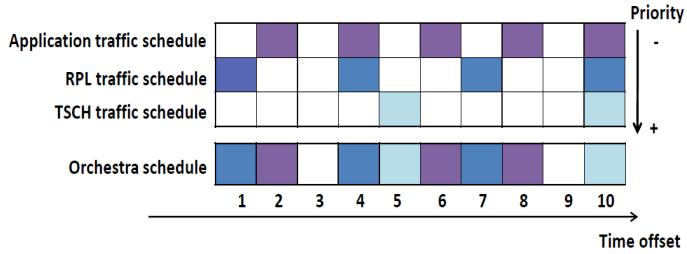


Fig. 1. Orchestra slotframes illustration (the slotframes lengths are 2, 3 and 5 corresponding to the application, routing and MAC slotframes respectively).

Each slotframe is composed of slots that can be dedicated (i.e. contention-free) or shared (i.e. contention-based with CSMA back-off). The slot coordinates (time and channel offsets) can be either fixed or variable —a function of the receiver/sender identifier. Generally, Orchestra identifies the following four types of slots:

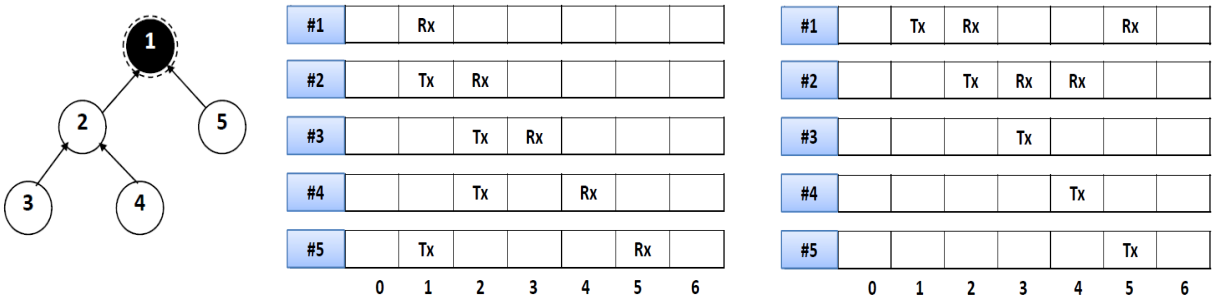
- Common Shared (CS) Slots: installed at fixed time and channel offsets. At every node, a CS slot results in a single shared slot used by all the nodes in the network for both transmission and reception.
- Receiver-based Shared (RBS) Slots: installed at coordinates that are derived from the identifier of the receiver node. Typically, the RBS are used for child-to-parent communication (i.e. application slotframe). At each node, RBS set-up corresponds to one reception (Rx) slot, which offset is computed as the node identifier modulo the slotframe length, and one transmission (Tx) slot per neighbor, which offset is computed as the neighbor identifier modulo the slotframe length. The node identifier should be unique, e.g. a node ID or a hash of the node's MAC address. Whenever switching parent, the child node updates its Tx slot autonomously and the parent node does not need to update its Rx slot. The RBS may arise contention since the parent node receives packets from all of its child nodes using one Rx slot. These child nodes use TSCH exponential back-off to contend for the channel and send their data. Figure 2a and 2b show an example of topology and the associated application schedule (made of RBS) of each node, respectively.
- Sender-based Shared (SBS) Slots: installed at coordinates derived from the identifier of sender node. At each node, SBS set-up corresponds to a Rx slot per neighbor where the offset is equal to the neighbor's identifier and one Tx slot which offset is equal to the node own identifier. When parent switching occurs, the old parent needs to remove the old Rx slot and the new parent requires to add a new Rx slot (a child node does not need to update its Tx slot). Compared to RBS slots, the SBS slots may decrease contention (as they avoid the installation of a Rx slot for all senders (child nodes), but also increase energy consumption as argued in [19]. Figure 2c illustrates an example of application schedule made of SBS slots.
- Sender-based Dedicated (SBD) Slots: slightly different from the SBS, in the sense that they use dedicated slots (instead of shared), and thus result in contention-free transmissions. They require that the slotframe length is long enough (longer than the number of nodes in the network) to install unique Tx slots to each node.

Each slotframe is established based on one of the slot types described above. The authors in [19] introduced an example of Orchestra setup consisting of three slotframes (Figure 1), described as follows:

- MAC slotframe (for TSCH traffic): made of SBD slots, so that TSCH beacon transmissions are contention-free. It has a length of 397 slots, which represents the maximum allowed number of nodes in the network. The highest priority is assigned to this slotframe.
- Routing slotframe (for RPL traffic): made of CS slots. Its default length is 31 slots. It has a lower priority than the MAC slotframe.
- Application slotframe: consists of RBS slots used for unicast data transmissions. Its length is fixed to 11 slots. Longer application slotframe leads to higher contention rates [19]. The application slotframe has the lowest priority.

In this paper, we adopted the above Orchestra setup introduced in [19]. Further, we used the implementation of TSCH and Orchestra[1] in Contiki. In this implementation, the number

[1]https://github.com/simonduq/orchestra



(a) A topology example (b) The associated schedule made of RBS slots (c) The associated schedule made of SBS slots

Fig. 2. Application schedule in Orchestra - illustration with a slotframe of length 7 time slots.

of packet retransmissions is set to 8 retries. Further, TSCH runs over the four best channels: 15, 20, 25, 26, where a channel offset is assigned to each slotframe (i.e. MAC, routing and application slotframes). In each slot, a node changes the communication channel, using the following inputs: the network's absolute slot number (ASN), the scheduled channel offset, and the hopping sequence ($\{15, 20, 25, 26\}$). Table I summarizes the scheduling rules used for time and channel offsets assignment.

TABLE I
TIME AND CHANNEL OFFSETS ASSIGNMENT RULES IN THE ORCHESTRA SCHEDULE [19].

Slotframe	Channel offset	Slot type	Slot offset
MAC slotframe	0	SBD	Tx offset = node ID, Rx offset = neighbor ID
Routing slotframe	1	CS	Tx offset = Rx offset = 0
Application slotframe	2	RBS	Rx offset = node ID % slotframe period, Tx offset = neighbor ID % slotframe period

III. RELATED WORK

Several scheduling algorithms have been proposed for TSCH networks. They can be broadly classified as either centralized [12–14] or distributed [15–18] algorithms.

An example of centralized algorithms is TASA (Traffic Aware Scheduling Algorithm) [12], which exploits the graph theory methods of matching and coloring to set up the schedule at the sink node.

MODESA (Multichannel Optimized Delay time Slot Assignment) [13] is another centralized scheduling solution that ensures a fair medium access adjusted according to the traffic demand of each node. It sorts the nodes according to their priorities, where highest priority corresponds to highest number of remaining packets in one node's buffer. Thus, MODESA first schedules the transmission of nodes having the highest priority, then iterates on the remaining set of nodes sorted based on their priorities.

MUSIKA (MUlti-SInK slot Assignment) [14] is an extension of MODESA [13] that considers multiple sink nodes and traffic differentiation. Each data packet has an importance degree (from the application point of view) and a destination sink. The authors used several sinks to associate each class of traffic to a specific sink. Traffic having the same importance degree and the same destination sink forms a traffic class and is pushed in a specific queue (i.e. each node maintains a FIFO queue per traffic class). Each node starts by transmitting packets in the queue of the traffic class with the highest importance degree.

Generally, centralized algorithms assume that a central entity has a full knowledge of the routing topology, the physical connectivity, and the traffic load of each node, which allows building then disseminating the schedule to all the nodes in the network. Unfortunately, centralized scheduling algorithms are not suitable for TSCH networks deployed in smart grid: first, they can present scalability issues, which is one of the requirements of WSN-based smart grid applications [21]. Second, in electric grid environments, typically characterized by highly corrosive conditions, routing topology and physical connectivity are highly dynamic and often unpredictable [21]. Consequently, any change in the network state should be communicated to the central entity to re-compute and redistribute new schedules, resulting in considerable signaling overhead.

Distributed scheduling algorithms have been proposed to resolve the above problems. DeTAS (Decentralized Traffic Aware Scheduling) [15, 22] is a distributed version of TASA that targets RPL networks with several sink nodes. The network is modeled as multiple routing graphs, each rooted at a different sink. For each graph, DeTAS builds a micro-schedule accommodating the transmissions of the nodes belonging to it. The micro-schedules are combined into a global macro-schedule.

Wave [16] is another example of distributed scheduling algorithms for TSCH, were the schedule is build by computing a series of waves. The sink node sends a message to its children to trigger the computation of the first wave. Each node having received the message selects a cell (a time slot and a channel) in the wave to transmit its first packet. Then, it notifies this assignment to its conflicting nodes. Once computed, the first wave constitutes the (time slot, channel) pattern: each successive wave is an optimized copy of the first wave, where the cells that are no more needed (i.e. that does

not contain transmissions) are removed.

Another distributed scheduling algorithm is proposed in [17], where the communication end-to-end delay is bounded by guaranteeing the delivery of each packet within a single slotframe. The network is organized into stratums, so that nodes located in a given depth have the same stratum. Each stratum is associated to a portion of the schedule (a block). To avoid funneling effects, the size of a block *i* is twice larger than the block *i+1*. When a node has many packets to forward (including re-transmissions), it negotiates with its parent to add slots using the 6top protocol [23].

In [18], the authors proposed a distributed TSCH schedule, based on the known PID (Proportional Integral and Derivative) control algorithm. The proposed schedule adjusts the number of allocated slots to the traffic demand and reacts to sudden traffic variations (such as bursty traffic) by adding/removing cells in the schedule using 6top protocol.

The OTF (On-the-Fly) bandwidth reservation [24] is another distributed scheduler that aims to dynamically adjust the node's schedule: based on the number of packets being sent to a given neighbor and the number of cells already reserved to that neighbor, OTF estimates the number of cells to be added/removed (to that neighbor). Then, it asks the 6top protocol to allocate/deallocate these cells, which triggers a negotiation process between neighbor nodes.

Generally, in distributed scheduling solutions, each node computes its local schedule based on information exchanged with its neighbors, such as topology and traffic information, in addition to scheduling information. However, distributed scheduling algorithms reduce, but do not eliminate the signaling overhead compared to centralized approaches.

Orchestra [19] is a recent scheduling technique that is neither centralized nor distributed (refer to Section II-C). Rather, it is referred to autonomous scheduling technique as each node autonomously builds its own schedule without any negotiation with its neighbors. The schedule is computed based on available routing information, using simple scheduling rules, and is automatically updated whenever the routing topology evolves. In terms of performance, Orchestra has been shown to provide high reliability but also high packet delays (as shown in the next section) because it does not account for traffic profiles. Accordingly, a new protocol will be designed in the rest of the paper to take advantage of Orchestra strengths and lift its restraints.

IV. Evidence of Orchestra limitation

In this section, we analyze the performance of TSCH based on Orchestra for schedule establishment, using COOJA [25] simulator, available as part of Contiki Operating System. Especially, we show that despite its outstanding features, Orchestra has the limitation of computing the TSCH schedule at each node, independently from its traffic load/demand, which can drastically affect the communication delay.

To this end, we have considered an RPL organized, TSCH-based network where nodes are deployed in a uniform grid topology. We varied the number of nodes from 49 (i.e. 7x7 grid) to 100 (i.e. 10x10 grid). The *DODAGroot* is located at coordinates (0,0) and the grid unit is equal to 30 meters. All the nodes in COOJA were configured as Tmote Sky motes whose transmission power is set to 31 dBm. The data rate is equal to 2 packets per minute. To enable the establishment of the topology, the nodes begin the transmission of data packets after a delay of 2 minutes. Each simulation run lasts 1 hour to ensure convergence to a steady state. Simulation results are provided with a 90% confidence interval.

To show the insensitivity of Orchestra to traffic demand, we investigate the performance of TSCH with Orchestra, under different traffic loads. Two different scenarios are considered:

- In the first scenario, only leaf nodes generate data packets and the rest of the nodes just forward these packets to the root node.
- In the second scenario, all the nodes are data sources except the root node. This scenario leads to congestion at nodes close to the root node termed as funneling effect [26]. Hence, this scenario allows analyzing the protocol performance when the traffic is not uniformly distributed in the network.

Figure 3 shows the average end-to-end delay of Orchestra in both scenarios. From this figure, it is clear that scenario 2 leads to a much higher communication delay compared to scenario 1. Further, in scenario 2 the delay increases like exponentially with the increase of number of nodes, in contrast to scenario 1 where the delay is almost constant. These results are justified as follows: in Orchestra schedule, each node reserves one Rx slot and one Tx slot per slotframe, independently from the amount of data traffic it has to deliver to the *DODAGroot*. However, this traffic load/demand varies according to the node location in the DODAG. For instance, nodes close to the root have higher traffic load than leaf nodes. Hence, in scenario 2, where all nodes are data sources, there was an accumulation of packets in the queues of nodes close to the root due to their high load. Consequently, each of these nodes required several slotframes (one slotframe provides only one Tx slot) to empty its queue. These phenomena lead to unwanted delays at these particular nodes, which negatively affects the overall communication delay. This delay is particularly unacceptable when the application has to deliver critical data such as alarms.

To confirm the above observations, the distribution of traffic loads in the network was analyzed. In order to reflect the traffic load at a particular node, we use the maximum number of packets in its queue (the peak transmission queue). Figure 4 illustrates the cumulative distribution function (CDF) of traffic loads in the networks for both scenarios. From this figure, it can be observed that in scenario 1, the queues are almost empty: the reserved slots are sufficient to send packets generated by leaf nodes without waiting in queues. However, in scenario 2, the number of packets in queues is between 0 and 14. Further, almost 5% of nodes — those close to the root — suffer from congestion (have more than 10 packets in their queues), which affects the end-to-end communication delay in the network. For instance, given a slotframe having 11 slots,

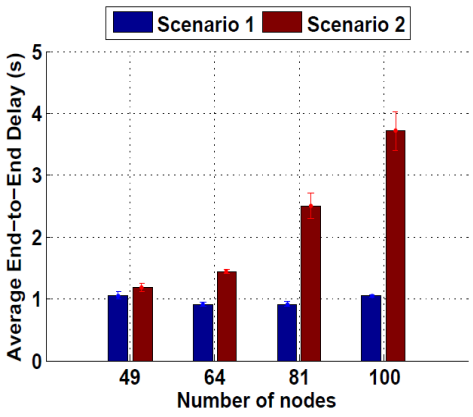


Fig. 3. End-to-end delay of Orchestra-based TSCH protocol, as a function of the number of nodes.

where the duration of a single slot is 15 ms, a node having 10 packets in its queue requires at least (without re-transmissions) 10 slotframes to empty its queue. Thus, the minimum delay to forward the packets in the queue is 1.65 s. This delay is definitely unacceptable for several critical applications such as the substation automation monitoring application, where sensed data must be received within a delay of 0.2 s in order to make the necessary protection actions [4].

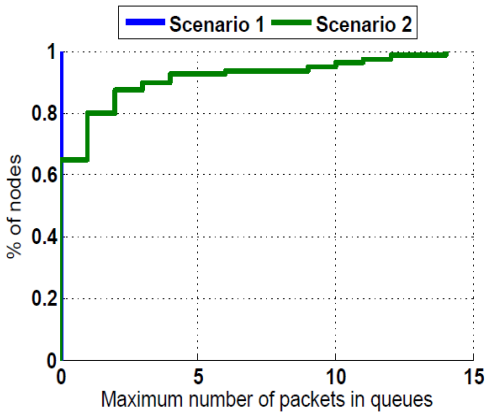


Fig. 4. Empirical CDFs of peak transmission queue — The number of nodes is fixed to 81.

V. E-TSCH-ORCH: ENHANCED ORCHESTRA-BASED TSCH PROTOCOL

In this section, our objective is to propose improvements for Orchestra-based TSCH protocol. The first improvement concerns the consideration of traffic load in scheduling to overcome the limitation of Orchestra discussed in the previous Section. The second proposed improvement concerns the support of traffic differentiation. The current TSCH specification handle all packets equally regardless of their criticality levels. However, WSN-based smart grid applications have different and often conflicting requirements in terms reliability and latency, resulting in heterogeneous data traffic patterns. Thus, we aim to enhance the TSCH protocol while introducing traffic differentiation.

Traffic load-based scheduling

To consider traffic load in scheduling, there are mainly two options. The first option is to assess traffic load at each node as the sum of traffic it generates and the traffic generated by the nodes belonging to its sub-tree, such as in [15, 16]. This solution leads to laborious neighbor-to-neighbor negotiation as each node has to collect information of traffic load from its children and then forwards the total traffic load information to its parent. The second option is to assess traffic load at each node as the number of pending packets in its queue, such as in [13, 18]. We adopt this solution for considering traffic load in Orchestra schedule, as it does not require any neighbor-to-neighbor negotiation.

Consider N_{rp}, the remaining number of packets to be transmitted, available in the node's waiting queue (buffer), and evaluated at each Tx slot offset. In Orchestra schedule establishment phase, each node computes one reception slot (Rx) to receive data in contention fashion from its children, and one transmission slot (Tx) to transmit its own data as well as that of its children towards its parent. To eliminate or at least reduce queued packets, we propose to add N_{rp} consecutive Tx slots to the schedule, starting from the Tx slot offset computed by Orchestra (refer to Table I). The number of added Tx slots is bounded by the slotframe size S: when the slotframe has sufficiently free time slots, i.e. $S >= N_{rp} + 1$, the node can exploit them to empty its queue; otherwise remaining packets in the node's queue, i.e. ($N_{rp} + 1 - S$) packets, are treated in the next slotframe. As each Tx slot should be synchronized with a Rx slot at the parent side, the number of added Tx slots is included in the footer of the transmitted packet to enable the parent node to increase/reduce the number of the Rx slots accordingly, by adding/removing the same number of consecutive Rx slots, starting from the Rx slot offset computed by Orchestra. Thus, adjusting (increasing or reducing) Tx and Rx slots dynamically according to each node traffic load and with Zero additional communication overhead can only improve the performance of TSCH based on Orchestra scheduling, especially in terms of latency.

Figure 5 represents an illustration of the proposed solution on the topology of Figure 2a. In this example, we have assumed that nodes 2 and 5 generate one packet every three slotframes (one packet per 495 ms), and nodes 3 and 4 generate one packet every two slotframes (one packet per 330 ms). We have chosen these values just as examples of data rates that make the illustration possible and easier. From Figure 5, we can see that packets are accumulated in the queue of node 2. For instance, at time offset 34, node 2 has transmitted one packet and has added two transmission slots (N_{rp} = 2) to empty its queue.

Traffic differentiation

In what concerns traffic differentiation, the second proposed enhancement for Orchestra-based TSCH, we consider two

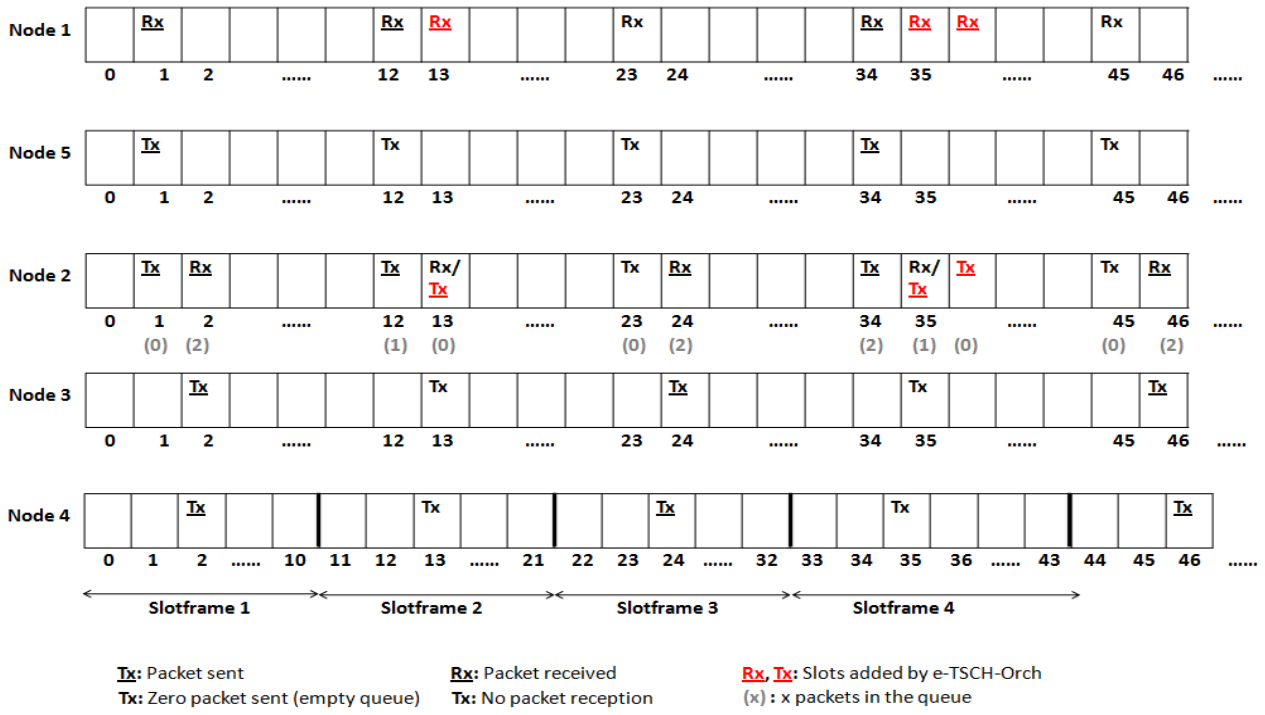


Fig. 5. Illustration of the proposed solution on the topology of Figure 2a.

traffic classes, periodic and critical. Periodic traffic, identified by low priority packets, is transmitted in a best-effort manner, while critical traffic (e.g. circuit breaker monitoring information and alert packet), identified by high priority packets, is favored over periodic traffic to ensure high reliability and short latency. In TSCH implementation under Contiki, the queue (buffer) is implemented as a ring, with a maximum of 16 packets in the queue, as illustrated in Figure 6. Therefore, to handle traffic differentiation, we propose to enqueue data packets differently, according to their priorities (as illustrated by Algorithm 1): critical traffic is pushed into the head of the queue while periodic traffic is pushed into the tail. By adopting this mechanism, critical traffic is transmitted before periodic packets and reaches the sink node within minimal delays. Such feature is of paramount importance for several delay-sensitive smart grid applications.

VI. Performance Analysis

In this section, we analyze the performance of the enhanced Orchestra-based TSCH protocol, denoted as e-TSCH-Orch, by comparing its performance against the original TSCH based on Orchestra scheduling, denoted as TSCH-Orch. Comparisons are carried out in terms of the following metrics:

- Average end-to-end delay: it represents the average duration separating a packet transmission by the sender and its reception by the *DODAGroot*.
- Average packet delivery ratio: it represents the ratio of the total number of received packets (at the *DODAGroot*) to the total number of sent packets (by all RPL router nodes), and is used as a measure of end-to-end reliability.

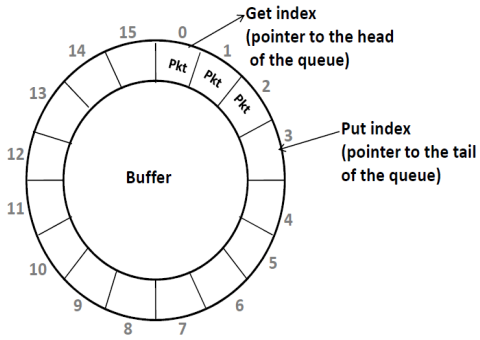


Fig. 6. The waiting queue (buffer) implemented as a ring.

- Average radio duty cycle: it represents the average portion of time spent with the radio on (either listening, receiving or transmitting), and is used as a measure of energy consumption.

For the performance evaluation of e-TSCH-Orch, we use two different environments. First, we use the COOJA simulator in order to conduct a massive simulation campaign and to be able to increase the number of nodes. Then, to obtain performance results with high confidence, we run experiments in a testbed of 20 Telosb nodes deployed in an office building. Next, we present the simulation and the experimental studies and their related results.

A. Simulation study

```
Initialization // the buffer is empty;
put_index ← 0;
get_index ← 0;
if the buffer is empty then
    insert_packet(put_index);
    increment(put_index);
else
    if the packet is critical then
        decrement(get_index);
        insert_packet(get_index);
    else
        insert_packet(put_index);
        increment(put_index);
    end
end
```

Algorithm 1: Adding a data packet to the node's queue

1) Simulation settings: We have considered the same simulation settings described in Section IV, where all nodes in the uniform grid topology are data sources except the root node. Further, we consider two traffic classes, periodic and critical with the respective proportions: 80% - 20%. As illustrated in Table II, parameters under consideration are number of nodes (49, 64, 81, 100), data rate (1 pkt/min, 2 pkts/min, 3 pkts/min, 4 pkts/min), and traffic heterogeneity (2 sorts of heterogeneous traffic and a homogeneous traffic). In what concerns the last considered parameter, traffic heterogeneity, note that homogeneous traffic occurs when all nodes communicate using the same data rate, namely 2 pkts/min, while heterogeneous traffic occurs when each node generates packets with different rates. As we considered two classes of packets i.e. periodic and critical whereby each class is generated with a different data rate, namely 1 and 3 pkt/min. Thus, we considered 2 sorts of heterogeneous traffic, referred as scenario 1 and scenario 2 as illustrated in Table III. Analyzing the impact of traffic heterogeneity, which is an outstanding feature of WSN-based smart grid applications, is of paramount importance for the experimentation of communication protocols in general and MAC protocols in particular. Next, we discuss results with respect to the described simulation scenarios.

TABLE II
SCENARIOS CONSIDERED IN OUR EVALUATION.

Scenario	Number of nodes	Data rate (pkt/min)	Traffic heterogeneity
1: Impact of number of nodes	49, 64, 81, 100	2	Homogeneous
2: Impact of data rates	64	1, 2, 3, 4	Homogeneous
3: Impact of traffic heterogeneity	64	1, 3	Heterogeneous, Homogeneous
Default settings	64	2	Homogeneous

2) Simulation results:

TABLE III
CONSIDERED SCENARIOS RELATED TO THE TRAFFIC HETEROGENEITY.

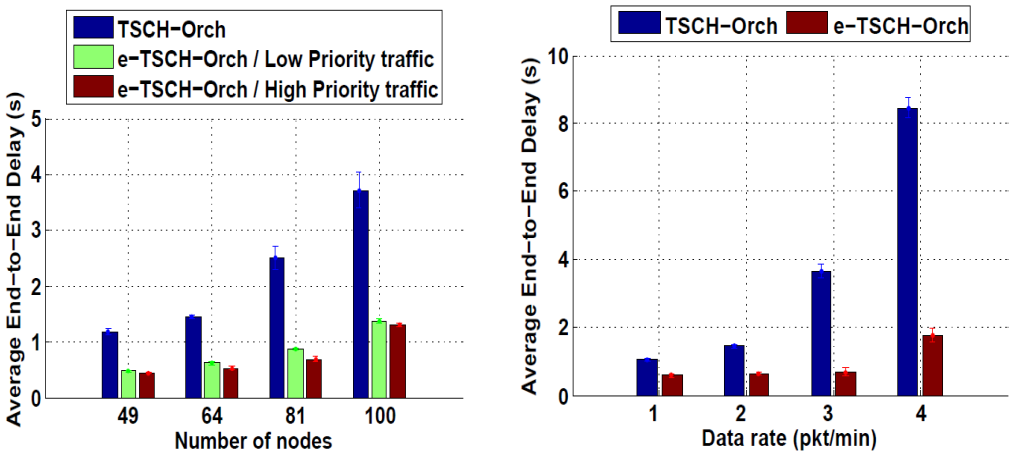
Scenario	Traffic pattern	Periodic Traffic data rate(pkt/min)	Critical Traffic data rate (pkt/min)
Scenario 1	Heterogeneous	3	1
Scenario 2	Heterogeneous	1	3
Scenario 3	Homogeneous	2	2

a) Average end-to-end delay: Figure 7a compares the average end-to-end delay of the initial and the enhanced protocols (TSCH-Orch and e-TSCH-Orch) as a function of the number of nodes. From this figure, we can retain two observations. First, Figure 7a illustrates the fact that e-TSCH-Orch delivers high priority packets before low priority packets. Thus, high priority packets always reach the root node within lower delays. As a second observation, e-TSCH-Orch improves the communication delay by 65.54% compared to TSCH-Orch. This observation can be justified as follows: the installation of additional communication slots in the current slotframe as a function of the traffic demand allows keeping the number of enqueued packets as small as possible and avoids waiting for upcoming slotframes for cleaning the queue. This observation can be also confirmed by the distribution of the traffic loads in the network, reflected by the maximum number of packets in queues, shown in Figure 8. In this figure, it can be observed that with e-TSCH-Orch, almost 99% of nodes has less than 4 packets in their queues. Further, the maximum number of enqueued packets drops to 6 packets, while it is equal to 14 packets with TSCH-Orch.

The average end-to-end delay of TSCH-Orch and e-TSCH-Orch as a function of the data rate is shown in Figure 7b. The results show that again, e-TSCH-Orch significantly improves the communication delay compared to the initial TSCH-Orch, which has been shown very sensitive to data rate variations. For example, for a data rate equal to 4 pkts/min, TSCH-Orch leads to an average delay greater than 8 seconds, which is considered to be too high. In contrast, e-TSCH-Orch is more robust to data rate variations. This is expected, since e-TSCH-Orch is designed to adjust the number of allocated slots with the traffic load (including traffic rate variations).

We have also explored the performance of e-TSCH-Orch in applications with heterogeneous and homogeneous traffic (refer to Section VI-A1 and Table III). Results are illustrated in Figure 9. We can observe that e-TSCH-Orch has a much lower communication delay than TSCH-Orch, in the three scenarios. Further, the delay of TSCH-Orch is significantly impacted by the heterogeneity of the data rates due to its constant transmission rate. In contrast, e-TSCH-Orch has almost the same communication delay, regardless of the considered scenario (traffic heterogeneity). This behavior is justified by the fact that e-TSCH-Orch allows each node to dynamically adjust its transmission rate according to the traffic load. Thus, e-TSCH-Orch can efficiently serves applications with heterogeneous data rates.

Notice that the increase in the number of communication slots in e-TSCH-Orch leads to an increase in the channel



(a) Delay as a function of the number of nodes.

(b) Delay as a function of the data rate.

Fig. 7. Average end-to-end delay of TSCH-Orch and e-TSCH-Orch.

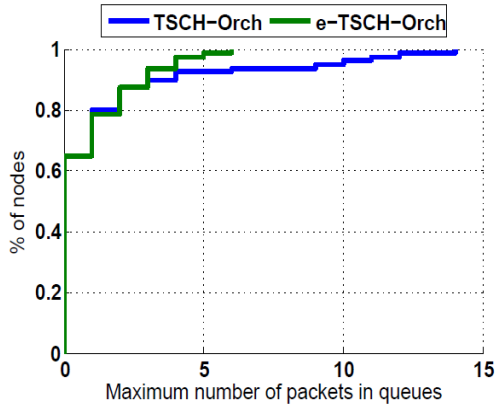


Fig. 8. Empirical CDFs of peak transmission queue — The number of nodes is fixed to 81 and the data rate is fixed to 2 pkts/min.

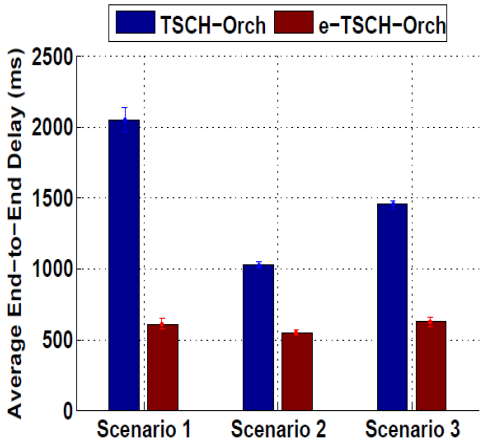


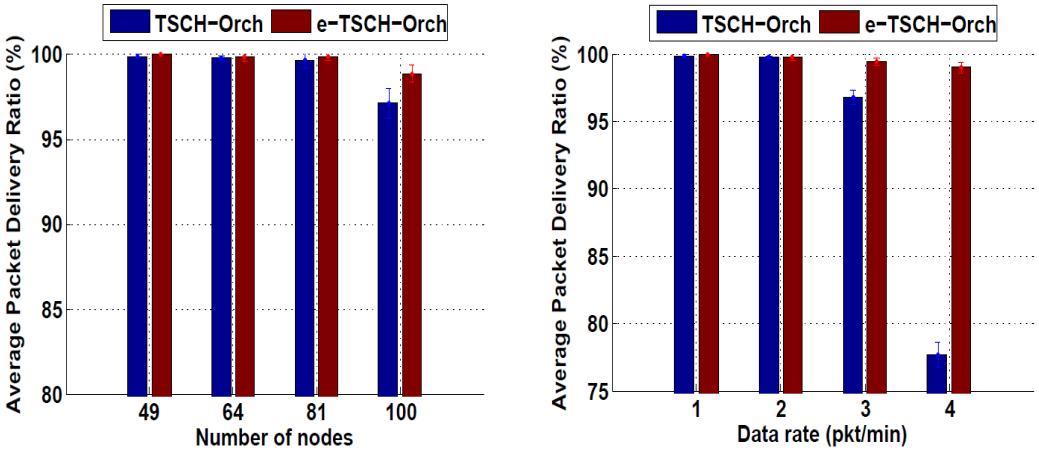
Fig. 9. Average end-to-end delay of TSCH-Orch and e-TSCH-Orch as a function of traffic heterogeneity — The number of nodes is fixed to 64.

hops, which induces a certain switching delay. However, in our study, we evaluated the end-to-end communication delay, derived as the average duration separating a packet transmission by the sender and its reception by the sink. This delay includes both channel switching and queuing delays. Analysis results demonstrate significant delay improvement of e-TSCH-Orch, over the original TSCH-Orch. Hence, we can conclude that although the delay induced by channel switching should be higher in e-TSCH-Orch than in TSCH-Orch, the overall end-to-end delay of e-TSCH-Orch is still significantly lower. This result is achieved thanks to the proper tuning of the time slots assignment according to traffic demand and criticality levels.

b) Average packet delivery ratio: Figure 10a depicts the average packet delivery ratio of TSCH-Orch and e-TSCH-Orch as a function of the number of nodes. For a network composed of less than 81 nodes, the two protocols have almost the same performance, which demonstrates that e-TSCH-Orch preserves the good performance of TSCH-Orch in terms of delivery. However, when the number of nodes is greater than 81, e-TSCH-Orch even improves the delivery over TSCH-Orch. This observation can be explained as follows: in TSCH-Orch, when the number of nodes increases, the funneling effect increases as well, leading to packets drops at congested nodes (buffer overflow). On the other hand, e-TSCH-Orch minimizes the number of enqueued packets and avoids buffers overflow, thus resulting in better delivery ratio.

Note that we have limited our study to a network size equal to 100 nodes, as we are interested in highly reliable networks. In fact, beyond 100 nodes, e.g. 121 nodes (11x11 grid), the packet delivery ratio (equal to 92%) decreases by 6.693% compared to the 100-nodes network. Thus, for a network consisting of more than 100 nodes, several RPL DAGs can be considered. In doing so, the proposed e-TSCH-Orch, applied to each DAG, can preserve its good performance in terms of delivery and delay as well.

In Figure 10b, the average packet delivery ratio of the



(a) PDR as a function of the number of nodes.

(b) PDR as a function of the data rate.

Fig. 10. Average packet delivery ratio of TSCH-Orch and e-TSCH-Orch.

original and the enhanced protocols as a function of the data rate are plotted. As it can be observed, the delivery ratio of TSCH-Orch drastically decreases when the data rate increases to 4 pkts/min, while e-TSCH-Orch keeps an almost constant delivery. This can be justified as follows: with the increase of the traffic rate, the congestion level of the network (specifically of nodes close to the root) increases, which leads to packets loss induced by buffer overflow. Since e-TSCH-Orch prevents (or reduces) the packets accumulation in queues (and thus buffer overflow), it preserves good delivery ratios even when the traffic rate increases to 4 pkts/min.

Beyond 4 pkts/min, simulation results (not included in Figure 10b) show a degradation in the network performance. For instance, using a network of 64 nodes (8x8 grid), the delivery ratio decreases from 99% for 4 pkts/min to 70% for 5 pkts/min. A such low delivery ratio is unacceptable, especially in smart grid applications where reliability represents a key requirement [4]. The observed performance degradation mainly pertains to the Orchestra scheduling approach. As a matter of fact, the application slotframe consists of RBS slots, and has a length limited to 11 slots. The authors in [19] argue that longer application slotframe leads to higher contention rates. On the other hand, when data rate increases (e.g. 5 pkts/min), the number of available slots in the slotframe can be insufficient for e-TSCH-Orch to allow a congested node transmitting all the packets in its buffer queue. Consequently, this buffer queue will grow leading to buffer overflow/packet drops and thus network performance degradation (e.g. packet delivery decrease).

In summary, so that the proposed e-TSCH-Orch operates efficiently and preserves the good network performance, we recommend considering data rates that do not exceed 4 pkts/min. Such data rates are commonly used in Low power and Lossy Networks [19, 27–29] (particularly, in WSN-based smart grid applications [27, 28]).

c) Average radio duty cycle: Figure 11 shows the average duty cycle of TSCH-Orch and e-TSCH-Orch as a function of the number of nodes and data rates. This figure shows that the communication delay and delivery improvements of e-TSCH-Orch shown above, are at the cost of higher duty cycle. Specifically, the addition of the transmission and reception slots in the communication schedule increases the portion of time spent with the radio on and thus increases the energy consumption, especially at nodes close to the *DODAGroot* (refer to Figure 12).

For instance, in Figure 11b, the increase of traffic rate, leading to additional communication slots in the schedules of congested nodes, results in higher energy consumption. However, this higher energy consumption of e-TSCH-Orch only concerns nodes close to the root (situated at less or equal to 4 hops from the root), as shown in Figure 12. It can be clearly observed that the average duty cycle of e-TSCH-Orch decreases significantly as the depth increases and becomes close to that of TSCH-Orch starting from 5 hops.

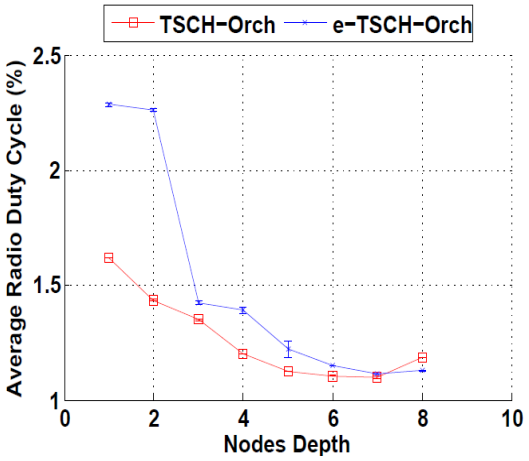
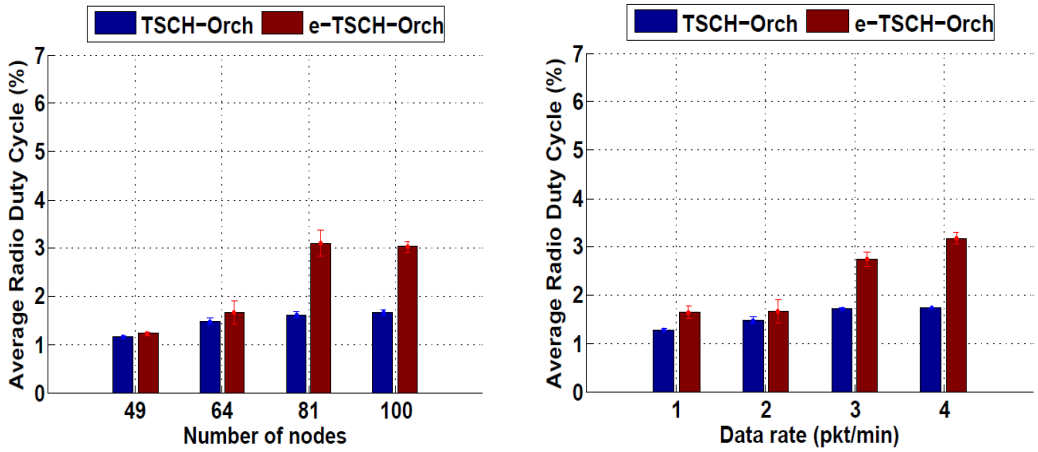


Fig. 12. Average duty cycle of TSCH-Orch and e-TSCH-Orch as a function of the nodes depth — The number of nodes is fixed to 64.



(a) RDC as a function of the number of nodes.

(b) RDC as a function of the data rate.

Fig. 11. Average duty cycle of TSCH-Orch and e-TSCH-Orch.

B. Experimental study

1) Experiments description: In our experiments, we have deployed a network composed of 20 Telosb motes (comprising a *DODAGroot* and 19 RPL routers) distributed in a tree topology. Figure 13 shows our experimental testbed, deployed in an indoor office environment. The transmission power of the Telosb motes is set to 3 dBm. For the fairness and brevity of our experimental study, we focus on scenario 2 (Table II in Section VI-A1), which consists in varying the data rate parameter. Each experiment lasted 30 minutes and repeated up to 5 times. Nodes began their transmission after a delay of 2 minutes to enable the topology establishment. In the remaining of this section, we present our experimental results, provided with a 90% confidence interval.

Fig. 13. Experimental testbed.

2) Experimental results: Figure 14 illustrates the average end-to-end delay of TSCH-Orch and e-TSCH-Orch protocols as a function of the data rate. It shows that e-TSCH-Orch achieves significantly lower end-to-end communication delays compared to TSCH-Orch, which is not robust to data rate variations. This observation confirms that found in the simulation study (Figure 7b). Moreover, it can be also confirmed from Figure 14 that data traffic is delivered according to the criticality levels: high priority traffic is always transmitted before low priority traffic.

In terms of average packet delivery ratio, it can be confirmed from Figure 15 that e-TSCH-Orch preserves its outperformance over TSCH-Orch. However, the good performance of e-TSCH-Orch in terms of delay and delivery leads to higher energy consumption: Figure 16 shows that e-TSCH-Orch increases the duty cycle of the network by 15.7% compared to TSCH-Orch. However, as explained in the simulation study, the duty cycle only increases for the nodes that are positioned close to the *DODAGroot* (congested nodes).

VII. Conclusion

In this paper, we have first conducted a performance analysis of Orchestra-based TSCH network. Performance analysis shows that the network may experience high end-to-end communication delay mainly because the schedule is not adaptive to traffic demand and some nodes suffer from high number of queued packets. Based on these results, we have designed a more efficient schedule, based on Orchestra that is aware of traffic demand. Depending on the number of packets a node still has in its buffer, each node can adaptively reserve additional transmission slots in order to keep a lower number of packets in the queue. In addition, we have improved the TSCH implementation in order to support different traffic types with different priorities and to favor the transmission of high priority traffic. The effectiveness of the proposed enhancements was thoroughly assessed through both simulations and real experiments. Analysis results demonstrate

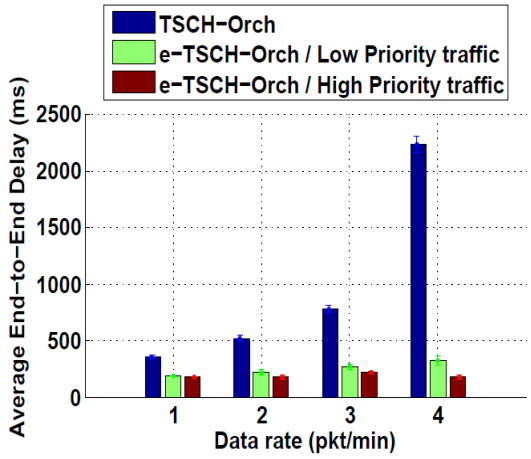


Fig. 14. Average end-to-end delay of TSCH-Orch and e-TSCH-Orch — experimental study.

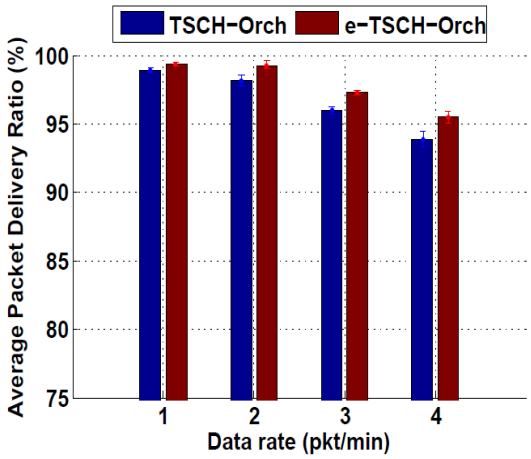


Fig. 15. Average packet delivery ratio of TSCH-Orch and e-TSCH-Orch — experimental study.

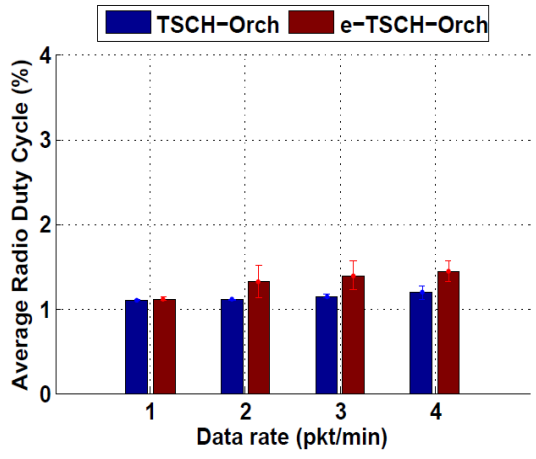


Fig. 16. Average duty cycle of TSCH-Orch and e-TSCH-Orch — experimental study.

that the proposed solution (i.) significantly reduces the end-to-end communication delay compared to the original Orchestra-based TSCH protocol, (ii.) preserves the packet delivery ratio, and (iii.) favors the delivery of high priority traffic. On the other hand, the enhanced Orchestra-based TSCH protocol shows higher average duty cycle due to the installation of additional communication slots. However, the increase in duty cycle only concerns nodes close to the root (congested nodes).

REFERENCES

[1] M. Erol-Kantarci and H. T. Mouftah, "Wireless multimedia sensor and actor networks for the next generation power grid," *Ad Hoc Networks*, vol. 9, no. 4, pp. 542–551, 2011. [Online]. Available: https://doi.org/10.1016/j.adhoc.2010.08.005

[2] E. A. Fadel, V. C. Gungor, L. Nassef, N. Akkari, M. G. A. Malik, S. Almasri, and I. F. Akyildiz, "A survey on wireless sensor networks for smart grid," *Computer Communications*, vol. 71, pp. 22–33, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2015.09.006

[3] V. C. Gungor, B. Lu, and G. P. Hancke, "Opportunities and challenges of wireless sensor networks in smart grid," *IEEE Trans. Industrial Electronics*, vol. 57, no. 10, pp. 3557–3564, 2010. [Online]. Available: https://doi.org/10.1109/TIE.2009.2039455

[4] V. C. Gungor, D. Sahin, T. Koçak, S. Ergüt, C. Buccella, C. Cecati, and G. P. Hancke, "A survey on smart grid potential applications and communication requirements," *IEEE Trans. Industrial Informatics*, vol. 9, no. 1, pp. 28–42, 2013. [Online]. Available: https://doi.org/10.1109/TII.2012.2218253

[5] U. DOE, "Communications requirements of smart grid technologies," *US Department of Energy, Tech. Rep*, pp. 1–69, 2010.

[6] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on smart grid communication infrastructures: Motivations, requirements and challenges," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 1, pp. 5–20, 2013.

[7] W. Wang, Y. Xu, and M. Khanna, "A survey on the communication architectures in smart grid," *Computer Networks*, vol. 55, no. 15, pp. 3604–3629, 2011.

[8] WG802.15, "IEEE 802.15.4-2015 - IEEE standard for low-rate wireless networks," 2016.

[9] Z. Shelby, S. Chakrabarti, E. Nordmark, and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-power Wireless Personal Area Networks (6LoWPANs)," 2012.

[10] T. Winter, "RPL: IPv6 routing protocol for low-power and lossy networks," 2012.

[11] P. Thubert, "An architecture for IPv6 over the TSCH mode of IEEE 802.15.4, draft-ietf-6tisch-architecture-11," 2017.

[12] 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 *23rd IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC 2012, Sydney, Australia, September 9-12*, 2012, pp. 327–332. [Online]. Available: http://dx.doi.org/10.1109/PIMRC.2012.6362805

[13] R. Soua, P. Minet, and E. Livolant, "MODESA: an optimized multichannel slot assignment for raw data convergecast in wireless sensor networks," in *31st IEEE International Performance Computing and Communications Conference, IPCCC 2012, Austin, TX, USA, December 1-3*, 2012, pp. 91–100. [Online]. Available: http://dx.doi.org/10.1109/PCCC.2012.6407742

[14] R. Soua, E. Livolant, and P. Minet, "MUSIKA: A multichannel multi-sink data gathering algorithm in wireless sensor networks," in *9th International Wireless Communications and Mobile Computing Conference, IWCMC 2013, Sardinia, Italy, July 1-5*, 2013, pp. 1370–1375. [Online]. Available: http://dx.doi.org/10.1109/IWCMC.2013.6583756

[15] 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 *IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks", WoWMoM 2013, Madrid, Spain, June 4-7*, 2013, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/WoWMoM.2013.6583485

[16] R. Soua, P. Minet, and E. Livolant, "Wave: a distributed scheduling algorithm for convergecast in IEEE 802.15.4e TSCH networks," *Transactions on Emerging Telecommunications Technologies*, vol. 27, no. 4, pp. 557–575, 2016. [Online]. Available: http://dx.doi.org/10.1002/ett.2991

[17] I. Hosni, F. Theoleyre, and N. Hamdi, "Localized scheduling for end-to-end delay constrained low power lossy networks with 6tisch," in *21st IEEE Symposium on Computers and Communication, ISCC 2016, Messina, Italy, June 27-30*, 2016, pp. 507–512. [Online]. Available: http://dx.doi.org/10.1109/ISCC.2016.7543789

[18] M. Domingo-Prieto, T. Chang, X. Vilajosana, and T. Watteyne, "Distributed PID-based scheduling for 6tisch networks," *IEEE Communications Letters*, vol. 20, no. 5, pp. 1006–1009, 2016. [Online]. Available: http://dx.doi.org/10.1109/LCOMM.2016.2546880

[19] S. Duquennoy, B. A. Nahas, O. Landsiedel, and T. Watteyne, "Orchestra: Robust mesh networks through autonomously scheduled TSCH," in *13th ACM Conference on Embedded Networked Sensor Systems, SenSys 2015, Seoul, South Korea, November 1-4*, 2015, pp. 337–350. [Online]. Available: http://doi.acm.org/10.1145/2809695.2809714

[20] S. Rekik, N. Baccour, M. Jmaiel, and K. Drira, "Holistic link quality estimation-based routing metric for RPL networks in smart grids," in *27th IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communications, PIMRC 2016, Valencia, Spain, September 4-8*, 2016, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/PIMRC.2016.7794925

[21] ——, "Wireless sensor network based smart grid communications: Challenges, protocol optimizations, and validation platforms," *Wireless Personal Communications*, 2017. [Online]. Available: http://dx.doi.org/10.1007/s11277-017-4038-1

[22] N. Accettura, E. Vogli, M. R. Palattella, L. A. Grieco, G. Boggia, and M. Dohler, "Decentralized traffic aware scheduling in 6tisch networks: Design and experimental evaluation," *IEEE Internet of Things Journal*, vol. 2, no. 6, pp. 455–470, 2015. [Online]. Available: http://dx.doi.org/10.1109/JIOT.2015.2476915

[23] Q. Wang and X. Vilajosana, "6tisch operation sublayer (6top) interface. draft," *IETF, July*, 2015.

[24] M. R. Palattella, T. Watteyne, Q. Wang, K. Muraoka, N. Accettura, D. Dujovne, L. A. Grieco, and T. Engel, "On-the-fly bandwidth reservation for 6tisch wireless industrial networks," *IEEE Sensors Journal*, vol. 16, no. 2, pp. 550–560, 2016. [Online]. Available: http://doi.org/10.1109/JSEN.2015.2480886

[25] F. Österlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with COOJA," in *LCN 2006, The 31st Annual IEEE Conference on Local Computer Networks, Tampa, Florida, USA, 14-16 November 2006*, 2006, pp. 641–648. [Online]. Available: https://doi.org/10.1109/LCN.2006.322172

[26] C. Wan, S. B. Eisenman, A. T. Campbell, and J. Crowcroft, "Siphon: overload traffic management using multi-radio virtual sinks in sensor networks," in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems, SenSys 2005, San Diego, California, USA, November 2-4, 2005*, 2005, pp. 116–129. [Online]. Available: http://doi.acm.org/10.1145/1098918.1098931

[27] E. Ancillotti, R. Bruno, and M. Conti, "The role of the RPL routing protocol for smart grid communications," *IEEE Communications Magazine*, vol. 51, no. 1, pp. 75–83, 2013. [Online]. Available: https://doi.org/10.1109/MCOM.2013.6400442

[28] ——, "Reliable data delivery with the IETF routing protocol for low-power and lossy networks," *IEEE Trans. Industrial Informatics*, vol. 10, no. 3, pp. 1864–1877, 2014. [Online]. Available: https://doi.org/10.1109/TII.2014.2332117

[29] O. Gaddour, A. Koubâa, and M. Abid, "Quality-of-service aware routing for static and mobile ipv6-based low-power and lossy sensor networks using RPL," *Ad Hoc Networks*, vol. 33, pp. 233–256, 2015. [Online]. Available: https://doi.org/10.1016/j.adhoc.2015.05.009