

Time-Critical Communication in 6TiSCH Networks

Abdulkadir Karaagac, Jetmir Haxhibeqiri, Ingrid Moerman, Jeroen Hoebeke
Ghent University - imec, IDLab, Department of Information Technology
Technologiepark Zwijnaarde 15, B-9052 Ghent, Belgium
{abdulkadir.karaagac, jetmir.haxhibeqiri, ingrid.moerman, jeroen.hoebeke}@ugent.be

Abstract—In Industry, there are number of time-critical applications with strong requirements for network services with latency and reliability guarantees. However, the lack of open and flexible, but also reliable and deterministic wireless communication solutions hinders the realization of fully connected and digital industrial spaces. In this paper, we investigate the approach of 6TiSCH architecture for deterministic networking performance and highlight its potential to be used in time-critical applications based on an experimental study.

Index Terms—6TiSCH, IEEE 802.15.4e TSCH, Time-Critical Applications, Industrial IoT, QoS

I. INTRODUCTION

Over the past few years, we have seen the uptake of connected solutions in industry to support and improve the operational performance in manufacturing, warehousing and distribution facilities. However, deterministic and reliable wireless communication in large-scale and dynamic applications still remains one of the greatest challenges for the realization of fully connected and digital industrial spaces.

From an industrial perspective, timely handling of critical events is vital so that system failures can be avoided or at least system down-time can be minimized. Particularly, in time-critical applications with strict real-time constraints, even a fractional violation of designated constraints can very well lead to performance drops or even system outages which can create significant cost for the industrial companies [1]. Beyond the classical case of industrial automation and control systems, there are in fact number of industries with strong and similar needs for deterministic and real-time network services with latency and reliability guarantees [2]. Currently, industry uses the combination of wired and wireless networking solutions. The wireless technologies are generally used for performing non-critical and monitoring activities, whereas wired infrastructure (i.e. Fieldbus, Industrial Ethernet) is used for time-critical or safety monitoring and control loop operations [3].

In this context, several wireless standards, such as WirelessHart [4] and ISA100.11a [5], were proposed and have been used to meet strong requirements of industrial applications. Although they have played a key role in industrial networks over the past decade, these widely used wireless standards turned out to be not always sufficient to meet the highly dynamic and strong expectations of today's industrial settings [6]. On one side they typically have difficulty coping with dynamic and mobile large-scale networks due to their network management scheme [7] and, on the other side they suffer limited flexibility due to not embracing ultimate digital Internet technologies.

Recently released amendment of IEEE 802.15.4, named “IEEE 802.15.4e”, offers a Time Slotted Channel Hopping (TSCH) mode which uses timely synchronized communication together with channel hopping to eliminate the impact of channel fading and interference [8]. Recent studies show that IEEE 802.15.4e TSCH has potential to ensure industrial-grade deterministic properties for applications with low latency, low jitter, low power and high reliability requirements [9].

To investigate IPv6 connectivity over the TSCH mode of IEEE 802.15.4e protocol, a new working group (WG), named 6TiSCH, has been recently formed within the IETF. It aims to link IEEE802.15.4e TSCH capabilities with the prior IPv6-enabled standards (IETF 6LoWPAN, RPL, CoAP) and recommendations. In addition to open standards-based architecture, 6TiSCH is also developing currently missing components to achieve industrial-grade performance for IPv6 over IEEE802.15.4e TSCH, such as schedule management, deterministic IPv6 flows, link management and routing [10].

With open and standardized communication stack and the support for different scheduling schemes and the concept of Deterministic Paths and Flow priorities, 6TiSCH Networks offer open and flexible, but also equally reliable and deterministic wireless communication solutions that can be used in time critical industrial applications. We believe by enabling a mix of a deterministic and time-critical connectivity with dynamic operations in scalable deployments, 6TiSCH is going to provide wireless access to a broader range of devices (sensors, actuators, wearables, etc.) and foster the development of numerous new applications and services in industrial domain.

This paper provides an overview of how 6TiSCH architecture is targeting deterministic and latency-bounded performance for end-to-end communication and also investigates the potential of 6TiSCH Networks to become one of the de-facto communication technologies for industrial applications with time-critical constraints. For that purpose, we present an experimental study which represents the deterministic behavior in 6TiSCH networks. We also detail and discuss about the missing points that should be targeted by the 6TiSCH.

The remainder of the paper is organized as follows. Section II provides detailed background on the target technologies in order to understand the current state of the art. Section III describes the aspects and mechanisms of 6TiSCH which contribute to the deterministic and latency bounded performance, whereas Section IV illustrates an experimental study on time-critical operation of 6TiSCH Networks. Further discussion is provided in Section V. Finally, Section VI concludes the paper.

II. TECHNICAL OVERVIEW

A. IEEE 802.15.4e TSCH

The IEEE 802.15.4e protocol is a recent MAC amendment of the IEEE 802.15.4 standard, specially designed for harsh industrial environments with a reliable and deterministic communication scheme based on Time-Slotted Channel Hopping [8]. The use of TSCH in industrial context is not a new approach, however IEEE 802.15.4e standard enabled TSCH networks for the standardized open connectivity.

In a TSCH network, time is sliced up into time slots and all motes across a network are synchronized in the order of tens of microseconds. Moreover, overall communication is orchestrated by a schedule which defines the action (transmit, receive, sleep) of each node in each time slot [8]. An example of TSCH schedule is provided in Figure 1. A single element, named 'cell', in the TSCH schedule is identified by a pair of slotOffset and channelOffset, which are used to define the time and frequency that will be used by communicating motes. By means of the combination of time synchronization and channel hopping, IEEE 802.15.4e is able to provide reliable and efficient communication [9].



Fig. 1: Example of TSCH schedule.

B. 6TiSCH: IPv6 over IEEE 802.15.4e TSCH

Recently, an IETF working group, named 6TiSCH, has been formed to investigate IPv6 connectivity over the TSCH mode of IEEE 802.15.4e protocol [10]. Currently, 6TiSCH WG is still active with two finalized RFCs and several active Internet Drafts. The working group has defined an operation sub-layer, called 6top, in order to bind IETF upper layers and IEEE 802.15.4e TSCH and it targets to create standard approach to build and maintain a schedule, perform TSCH configuration and control procedures. Also a minimal 6TiSCH configuration is defined in order to achieve basic interoperability among all implementations. The group is also working on a set of protocols for setting up a TSCH schedule in distributed approach with various scheduling functions [11].

The IETF 6TiSCH stack encompasses an Internet-enabled IPv6 based upper stack (IETF 6LoWPAN, RPL and CoAP) with state-of-the-art low-power wireless mesh communication technologies [12]. The projected 6TiSCH protocol stack is provided in Figure 2.

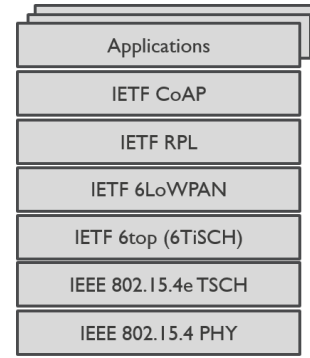


Fig. 2: 6TiSCH Protocol Stack.

C. Schedule Management in 6TiSCH Networks

The proper functioning of IEEE 802.15.4e TSCH depends on a schedule which can be typically created in various ways, but computed according to the specific requirements of the application, such as latency, reliability and energy consumption.

In 6TiSCH, there are 4 schemes defined to manage the TSCH schedule [10]. Firstly, there can be static schedules which are predefined or learned during the joining process, such as the shared slots that are used by every node in the network. Secondly, a remote schedule management scheme is defined where a central entity, called Path Computation Element (PCE), is continuously adjusting the TSCH schedule according to network state and traffic requirements of the motes in the network. Moreover, 6TiSCH also enables a Neighbor-to-Neighbor Scheduling which the motes agree on a schedule by using distributed multi-hop scheduling protocols and neighbor-to-neighbor scheduling negotiation (6P) [13]. Finally, a Hop-by-hop scheduling can be realized by a cell reservation along a path for a particular flow using a distributed mechanism. An overview of the 6TiSCH Network Architecture and Scheduling schemes are illustrated in Figure 3.

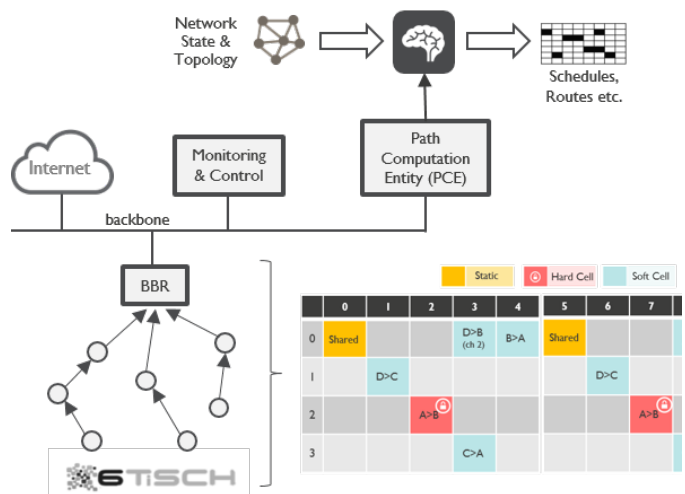


Fig. 3: Architecture and scheduling in 6TiSCH Networks.

As it is shown in the Figure 3, the cells scheduled via centralized scheduling are called “Hard Cell” and the 6top sublayer or any local entity cannot dynamically alter them. Whereas, a “Soft Cell” is scheduled by a distributed scheduling entity. An entity in the upper layer of 6top provides the number of the slots needed and 6top allocates the specific slots accordingly [13].

III. TIME-CRITICAL COMMUNICATION OVER 6TiSCH

In addition to the use of TSCH as medium access mechanism, which is already providing a level of determinism and reliability, the 6TiSCH architecture offers a suite of features for achieving industrial-grade deterministic performance (i.e. controlled latency and jitter, high reliability) for end-to-end communication.

A. Deterministic Networking: Centralized Scheduling

The Centralized Network and Schedule Management is one of the key functionalities of 6TiSCH to make wireless deterministic. As described in the Section II-C, the 6TiSCH allows using a centralized route and schedule management to achieve deterministic properties.

In this context, 6TiSCH performs the IETF DetNet Architecture, and IETF components such as the Patch Computation Element (PCE). So the PCE first collects network state information and traffic requirements from the motes across the network, and after obtaining a global view of the network, PCE build the schedules and routes by taking the QoS requirements of the all network. After the computation, the PCE installs the computed schedules to the related motes. Considering the dynamic and harsh environment in industrial spaces, the PCE needs to maintain scheduling process throughout all network operation and keep the schedules optimized according to any change in the network, environment or application requirements.

Considering the functionality of the centralized scheduling, PCE can perform optimizations based on the individual path delays. For instance, it can prioritize certain motes with time-critical data and provide minimized latency and redundancy cells for such nodes. In addition, centralized scheduling can reserve necessary resources for time-critical and sudden events that might be required during emergency states.

For the remote management of devices and schedules, the 6TiSCH architecture is adopting a common constrained device management interface from IETF CoRE Working Group, named the CoAP Management Interface (COMI) [14]. The CoMI protocol is based on CoAP [15] and CBOR [16] protocols and trying to define management operations with minimized load on the constrained devices.

B. Deterministic Flows: Tracks

A “Track” is, another concept instantiated by 6TiSCH architecture, a directed and deterministic path between source and destination node across a 6TiSCH network, in order to create guarantees of minimum and maximum end-to-end latency and also a tight jitter.

So, 6TiSCH establishes Tracks over the network for flows with specific requirements and it carries these flows with specific bounded latency and low jitter, without any influence of the network load or any kind of congestion and interference of other existing flows in the network [17]. 6TiSCH is achieving this guaranteed behavior by reserving channel resources (e.g communication schedules, memory buffers) at each hop on the Track and prioritizing flow traffic with respect to any other data traffic.

As described in [10], the Track is composed of bundles of cells with related schedules and logical relationships that binds source and destination of the flow. So, all the nodes along a Track know both incoming and outgoing cells which are associated to a certain Track and each receiving cell is uniquely paired to a transmitting cell. In addition, the Track and the associated transmit bundle can be uniquely identified by the IPv6 source address and a local RPLInstanceID in the packet. Therefore, all the packets carried in a 6TiSCH network must carry the Instance ID that identifies the 6TiSCH topology that is to be used for routing and forwarding that packet [10].

During forwarding process, as the intermediate nodes receive the incoming frame (whose destination MAC address is set to broadcast “0xFFFF”) on the specific receiving cell, the frames can be switched to next hop by the 6top, via associated transmitting cell, without any change in the MAC header. Therefore, the Tracks are creating a layer-2 forwarding state that can be used with any network layer protocol, but with a lower process delay and overhead compared to layer-3 forwarding schemes [10]. An overview of 6TiSCH Track operation is presented in Figure 4.

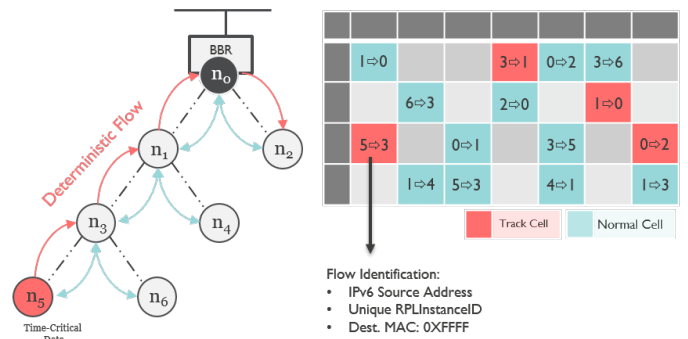


Fig. 4: 6TiSCH Track operation.

Another beneficial feature defined by the 6TiSCH architecture is the reuse of the reserved cells in order to avoid waste of resources. So that, any transmit cell associated with a Track can be opportunistically used by other routed packets, if the next-hop info matches and the particular cell is not currently used or under-utilized by the associated Track.

Concerning the creation and management of Tracks, 6TiSCH architecture originally allows both remote and distributed track management. However, at the time of writing, there is no specification or recommendation about the details of the Track computation, creation and management.

IV. EXPERIMENTAL EVALUATION

In order to illustrate the contribution of the mentioned mechanisms into the deterministic and latency-bounded behavior of 6TiSCH networks, we performed this experimental study. In the following sections, firstly, the experimental setup is explained. Then, a detailed description is provided for the experimentation scenarios, followed by the obtained results.

A. Experimental Setup

In these experiments, the OpenWSN OS [18], which is an open-source and reference implementation for the 6TiSCH networking stack, is used. Currently, this platform includes TSCH, and some of 6TiSCH related components (6top, SF0) and relatively simple implementations of upper IETF stack (RPL, 6lowpan and CoAP) [18]. However, currently OpenWSN does not implement any interface for remote network and schedule management and it does not support deterministic flow operations as defined in the 6TiSCH architecture.

Therefore, in order to create the behavior of Centralized Scheduling and Deterministic Flows, we had to tweak the OpenWSN code. For producing the centralized scheduling mechanism, we manually installed pre-calculated and optimized schedules on each mote remotely. This operation created a number of Hard Cells on the nodes, which cannot be altered by 6top, but can be used for any type of application traffic. For deterministic flows, we manually created bundles of Hard Cells on the relevant nodes and limit their usage for a certain data flow in order to mimic the Track behavior in 6TiSCH.

The testbeds are deployed on the OpenMote-cc2538 motes in an indoor environment with network topology provided in Figure 5. As it is shown in this figure, we used 1 Backbone Router and 6 sensor nodes; 4 monitoring nodes (n_1 , n_3 , n_4 and n_6) which can generate periodical non-critical data and 2 motes as source (n_5) and destination (n_2) of critical application data. Such heterogeneous network structures can be encountered commonly in the industrial setups. For instance, monitoring data might represent temperature or humidity monitoring of an insignificant process, whereas the critical data might be an alarm information that must be carried between an emergency button and a vital machine.

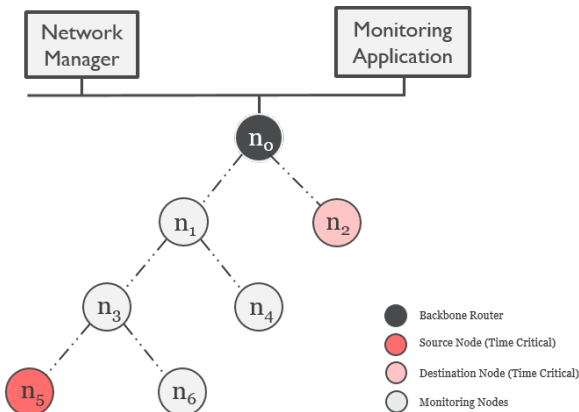


Fig. 5: The network topology for the experimental setup.

B. Experimentation Scenario

In order to analyze the contribution of centralized scheduling and track operations, we run experiments with a fixed scenario on 3 different communication scheme:

- Only Distributed Scheduling: In this scheme, the motes were only using 6top protocol, which was using Scheduling Function 0 (SF0) [11], in order to calculate their schedules in a distributed manner. The RPL protocol was used as the routing protocol.
- Combination of Distributed and Centralized Scheduling: The nodes were still using 6top and RPL protocol to construct the network. After the network initiation, we manually installed the pre-calculated optimized schedules on the motes where time-critical data traffic would occur, in order to improve the end-to-end performance.
- Deterministic Flows with Distributed Scheduling: In this scheme, we limited the use of hard cells to certain traffic flow for creating an deterministic flow for time-critical application data. Therefore, any monitoring data or other control messages will not be able to use these cells.

In each scheme, we first executed the network with only time-critical application traffic between source and destination node and collected end-to-end latency and jitter values for this traffic. Afterwards, we triggered the periodic generation of monitoring data by dedicated monitoring nodes, destined to a monitoring application in the backbone network. For both time critical and monitoring data, we used data generation period of 1 second. The detailed experimentation parameters and settings are summarized in Table I.

TABLE I: Detailed parameters and settings.

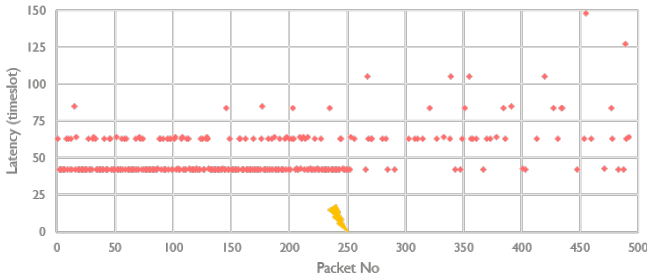
Parameter	Value
Number of Sensor Motes	6 + 1 BBR
Time-Critical Data Generation Period	1 second
Monitoring Data Generation Period	1 second
Queue Size	10
Time Slot Duration	10ms
Length of Slotframe	21
Number of Shared Slots	2
Number of Serial Slots	4
802.15.4 Channels	11 to 26
Maximum TSCH retries	3

C. Results

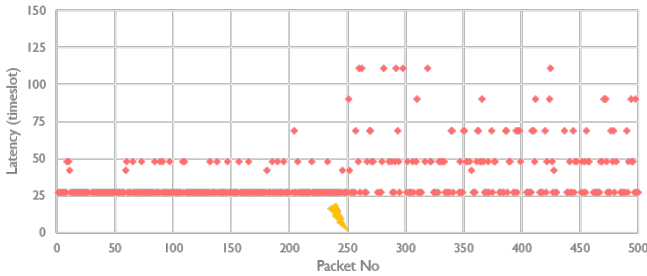
For each networking scheme, the obtained end-to-end latency values are presented in Figure 6. In these figures, the latency values for 500 time-critical packet are presented in terms of timeslots and the lightning sign represents the initiation of monitoring data after the 250th packet. The average latency and jitter values obtained during these experiments are also provided in Table II. In this table, the jitter values represents the average variation in the end-to-end delivery latency with respect to minimum delay in terms of timeslots.

The performance trends in the Figure 6a and in Table II show that the network with only distributed scheduling was performing relatively worse in terms of latency, determinism and reliability performance. Since the schedules are calculated locally by the neighbor nodes without the knowledge of overall network state, the minimum and average latency for any kind of data is relatively higher than other scenarios. Especially after the initiation of the monitoring data, we encounter a very high deviation in end-to-end latency and even a remarkable amount of packet loss for time-critical data.

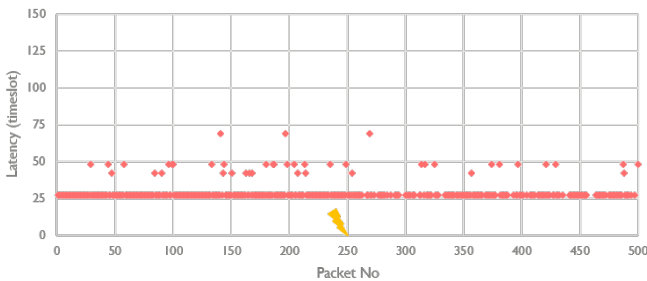
In the second scheme, where we installed schedules with optimized path delays, the minimum and average end-to-end latencies have already improved significantly compared to the distributed scheduling case. As it is presented in Figure 6b, the network was able to perform very well in terms of time-bounded packet delivery until the initiation of monitoring data. The results in Table II also verify that the network was even able to perform as good as deterministic flows when there was no monitoring traffic. However, with the insertion of monitoring data, the time-critical traffic faced a performance drop and consequently the latency and jitter of the packets has escalated.



(a) Only Distributed Scheduling.



(b) Combination of Distributed and Centralized Scheduling.



(c) Deterministic Flows with Distributed Scheduling.

Fig. 6: End-to-end latency values for time-critical data.

Concerning the last scheme, where we created dedicated and deterministic flows for time-critical data, the network was able to perform very well throughout the whole experiment. As it is illustrated in Figure 6c and Table II, the quality of service for the time-critical data traffic was not influenced by the non-critical data communication. Therefore, the network was still able to provide latency-bounded and reliable communication for time-critical traffic even with other inserted traffic across the network.

TABLE II: Average end-to-end latency and jitter values for time-critical data traffic.

Measurement Scenario	Delay (ts)	Jitter (ts)
Distributed Scheduling - No Monitoring	48.68	6.68
Distributed Scheduling - With Monitoring	107.4	65.4
Cent. and Dist. Scheduling - No Monitoring	29.3	2.3
Cent. and Dist. Scheduling - With Monitoring	62.54	35.54
Det. Flows and Dist. Sch.- No Monitoring	29.055	2.055
Det. Flows and Dist. Sch, - With Monitoring	29.289	2.289

These experiments show that the centralized scheduling can provide best-performance schedules and improve the latency and reliability in the 6TiSCH networks, compared to the distributed scheduling. However, it cannot provide any performance guarantee or time boundary as always desired for time-critical applications. Therefore, deterministic flows with dedicated cells and reserved resources are crucial and still required in order to create networks with deterministic behavior without any influence of the network load due to other network traffic.

V. DISCUSSION AND FUTURE WORK

With open and standardized communication stack and the support for different scheduling schemes and the concepts of Deterministic Paths and Flow priorities, 6TiSCH Networks has potential to offer open and flexible, but also equally reliable and deterministic wireless communication solutions that can be used in time critical industrial applications. Although, the 6TiSCH architecture is offering all these features for achieving industrial-grade deterministic performance for end-to-end communication, the standardization is not complete yet and there are several issues which require further attention from research community.

Currently, the main focus of the 6TiSCH community is on the 6top protocol and distributed scheduling mechanisms. Although the functionalities about the centralized scheduling and deterministic flows are defined in the 6TiSCH architecture, the details about their creation, installation and management are not standardized or described yet. Therefore, as a first step, the design and standardization of the adopted 6TiSCH remote management interfaces and their functionalities must be finalized.

Despite the fact that distributed schedule management is more favorable candidate for highly dynamic networks, the centralized approach is more preferable for the stable networks via offered optimal performance [19]. However, the remote management operations might require more signaling overhead in order to maintain the schedules across the network. These control message overheads lead to concerns about the scalability of such networks and their responsiveness against dynamic settings. Therefore, the cost of control messages on the network resources (especially aiming the resource efficiency of CoMI interfaces) must be studied and the limitations and capabilities of centrally managed 6TiSCH networks must be clarified.

The 6TiSCH supports multiple methods (i.e. static, remote, neighbor-to-neighbor, hop-by-hop) to manage schedules and to reserve network resources. Therefore, it is possible to run distributed scheduling over the network for all nodes with minimized overhead, and then create optimized and interference-free schedules only for time-critical data when it is necessary. This approach can enable a mix of a deterministic and time-critical connectivity with dynamic operations in scalable deployments. However, the coordination and integration of centralized and distributed management schemes for 6TiSCH nodes is a novel issue that must be further investigated and standardized.

The proper functioning of IEEE 802.15.4e TSCH networks depends on communication schedules which can be typically generated in various ways, but must be computed according to the specific requirements of the target application, such as latency, reliability and energy consumption. However, looking at today's state-of-the-art research on scheduling, we see that it only makes simple assumptions on traffic requirements and it is incapable to automatically derive these application requirements. Therefore, there is a need for further investigation of application awareness and automatic application requirements determination (mainly focusing on 6TiSCH networks) in industrial applications.

Currently, the 6TiSCH protocol describes only the deterministic behavior locally in IEEE 802.15.4e Networks. However, for the purpose of fully integrated and digital industrial systems, it is also essential to achieve end-to-end deterministic communication between the constrained nodes in different Low Power and Lossy Networks (LLNs) or other entities in the wide area networks. Therefore, the preservation of deterministic behavior in the backbone network must be also addressed by the research community in conjunction with 6TiSCH architecture.

As a future work, we would like to investigate aforementioned research points and contribute to the standardization and development of 6TiSCH Networks, starting from the design and implementation of remote management interfaces. Then, we will implement all of the functionalities, concerning deterministic behavior, defined in the 6TiSCH protocol, in OpenWSN OS, and try to realize and demonstrate open and standard-based deterministic wireless communication for time-critical industrial applications.

VI. CONCLUSION

This paper investigates the 6TiSCH approach for the deterministic networking behavior targeting time-critical applications, specifically targeting industrial systems. In detail, centralized scheduling and deterministic flow mechanisms are examined first and their possible contribution to the deterministic network behavior, in terms of end-to-end latency and jitter, is presented by means of a suite of experiments. However, due to the fact that the 6TiSCH architecture is still in design and development phase, there are still severe points in 6TiSCH protocol which need to be further investigated by the research community.

REFERENCES

- [1] T. Sauter, "Accessing Factory Floor Data", in R. Zurawski (Eds.), *Industrial Communication Technology Handbook*, 2nd Ed., vol. 8, no. 2, pp. 9-10, June 2014.
- [2] E. Grossman et al., "Deterministic Networking Use Cases", IETF Standard draft-ietf-detnet-use-cases-12, April 2017.
- [3] D. Greenfield, "Industrial Networks: Wired & Wireless", *Automation World*, November 2013.
- [4] IEC-62591, "Industrial communication networks - Wireless communication network and communication profiles-WirelessHART".
- [5] IEC-62734, "Industrial networks - Wireless communication network and communication profiles - ISA 100.11a".
- [6] P. Zand, S. Chatterjea, K. Das, P. Havinga, "Wireless Industrial Monitoring and Control Networks: The Journey So Far and the Road Ahead", *Journal of Sensor and Actuator Networks*, 2012.
- [7] P. Zand, S. Chatterjea, J. Ketema, P. Havinga, "A Distributed Scheduling Algorithm for Real-time (D-SAR) Industrial Wireless Sensor and Actuator Networks, ETFA 2012.
- [8] "IEEE std. 802.15.4e, Part. 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer", IEEE Computer Society, 2012.
- [9] M.R. Palattella, P. Thubert, X. Vilajosana, T. Watteyne, Q. Wang, and T. Engel, "6TiSCH Wireless Industrial Networks: Determinism Meets IPv6," in Mukhopadhyay S. (eds) *Internet of Things. Smart Sensors, Measurement and Instrumentation*, vol 9. Springer, Cham, 2014.
- [10] P. Thubert, "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4", IETF Standard draft-ietf-6tisch-architecture-12, August 2017.
- [11] D. Dujovne, L. A. Grieco, M. R. Palattella, N. Accettura, "6TiSCH 6top Scheduling Function Zero / Experimental (SFX)", IETF Standard draft-ietf-6tisch-6top-sfx-00, September 2017.
- [12] T. Watteyne, P. Tuset-Peiro, X. Vilajosana, S. Pollin and B. Krishnamachari, "Teaching Communication Technologies and Standards for the Industrial IoT? Use 6TiSCH!," in *IEEE Communications Magazine*, vol. 55, no. 5, pp. 132-137, May 2017.
- [13] Q. Wang, X. Vilajosana, T. Watteyne, "6top Protocol (6P)", IETF Internet-Draft draft-ietf-6tisch-6top-protocol-09, October 2017.
- [14] M. Veillette, P. Van der Stok, A. Pelov, A. Bierman, "CoAP Management Interface", IETF Internet-Draft, draft-ietf-core-comi-01, July 2017.
- [15] Z. Shelby, K. Hartke, C. Bormann, "The Constrained Application Protocol (CoAP)", IETF RFC 7252, June 2014.
- [16] C. Bormann, P. E. Hoffman, "Concise Binary Object Representation (CBOR)", IETF RFC 7049, October 2013.
- [17] P. Thubert, M. R. Palattella and T. Engel, "6TiSCH centralized scheduling: When SDN meet IoT," 2015 IEEE Conference on Standards for Communications and Networking (CSCN), Tokyo, 2015, pp. 42-47.
- [18] T. Watteyne, X. Vilajosana, B. Kerkez, F. Chraim, K. Weekly, Q. Wang, S. Glaser, K. Pister, "OpenWSN: A Standards-Based Low-Power Wireless Development Environment", *Wiley's Transactions on Emerging Telecommunications Technologies*. Volume: 23: Issue 5. 480-493. August 2012.
- [19] T. P. Duy, T. Dinh, Y. Kim, "Distributed cell selection for scheduling function in 6TiSCH networks", In *Computer Standards & Interfaces*, Volume 53, 2017, Pages 80-88.