

# 6TiSCH Centralized Scheduling: when SDN Meet IoT

Pascal Thubert\*, Maria Rita Palattella<sup>†</sup>, and Thomas Engel<sup>†</sup>

\**CISCO Systems, Sophia Antipolis (France), pthubert@cisco.com*

<sup>†</sup>*SnT, University of Luxembourg (Luxembourg), {maria-rita.palattella, thomas.engel}@uni.lu*

**Abstract**—The Deterministic Networking paradigm, which is prevalent in Operational Technology (OT), is now getting traction at the IETF and the IEEE, to enable the convergence of OT with Information Technology (IT), and the Industrial Internet vision, whereby the automation world can leverage IT technology to optimize OT processes. New Working Groups (WGs) are now emerging at the IETF to develop new routing and resource allocation schemes for IPv6/MPLS-based deterministic Layer-3 networks. In this work, we present the challenges that the DetNet and the 6TiSCH efforts will be facing to enable that convergence, particularly how DetNet can apply Software Defined Networking (SDN) centralized methods to provide global optimizations, and how 6TiSCH can reuse and extend the DetNet work for LowPower Wireless Sensor Networks (WSNs).

**Index Terms**—SDN, IoT, WSN, PCE, scheduling, deterministic networks, standardization, 6TiSCH, DetNet

## I. INTRODUCTION

The emergence of the Internet of Things (IoT) represents a brutal increase of new connected devices for which the current Internet infrastructure was not prepared; a new networking architecture is required to manage the *explosion* of IoT flows and allow the coexistence of different services with drastically different Quality of Service (QoS) requirements. The early adoption IoT was initially slowed by incompatible proprietary solutions that relied on specific gateways and maintained both OPEX and CAPEX high. As is often the case, standardizations bodies and industry associations moved in to develop standards that would restore the end-to-end principle, guarantee inter-operation between devices from different vendors, simplify deployments and lower costs [1]. At the IETF, various Working Groups (WGs), notably 6LoWPAN (today 6lo<sup>1</sup>) and ROLL<sup>2</sup>, introduced new protocols to integrate of low-power wireless networks into the Internet, by proposing mainly distributed solutions for address assignment and routing.

Despite this trend, nowadays, to manage large-scale IoT networks, establish complex routing topologies and simplify user operations in non-IT environments, the need of a *centralized network control* has emerged. Therefore, *Software Defined Networking* (SDN), by centralizing network control, and by providing a dynamic, flexible, and automated reconfiguration of the network, is foreseen as a *key* and *critical enabler* for the Internet of Things. Beside recent research activities on this topic [2]–[4], the SDN paradigm has evolved over the past last

20 years [5], and was already applied in early Wireless Sensor Networks (WSNs), which prefigured the IoT as we know it today. One of the main concept which has paved the way to SDN is the *separation of control and data plane*: this idea started gaining interest in the early 2000s with the proposition of traffic engineering (TE) techniques, aiming to control the paths used for delivering traffic, in order to better manage traffic and allocate resources, while providing reliable communications [5]. In line with this logically centralized control of the network, the Path Computation Element (PCE) protocol [6] was proposed at IETF. PCE, designed for MultiProtocol Label Switching (MPLS) networks, separates the route computations from the signaling of end-to-end connections and from actual packet forwarding, following the same logic of SDN-based networks.

The centralized routing model promoted by SDN has been adopted by ISA100.11a and wireless HART which are the major technologies which have been used so far for deploying industrial WSNs. These standards, which have been instrumental in the evolution of IoT [1], use a controller to compute all routes in the mesh network. These routes are generally multi-path, so as to augment the spatial diversity that is offered to the transported flows and to route around interferences dynamically.

Due to the need of a centralized computation to solve the NP-complete problem of multi-path route optimization, those networks do not generally scale to large configurations and are too costly to efficiently address large scale monitoring applications such as required for the future *Industrial Internet*.

With Industrial Internet we refer to the convergence of Operational Technology (OT) and Information Technology (IT), where deterministic industrial flows will be supported over traditional best-effort IP network architecture, which relies on selective queuing and discarding of packets to achieve end-to-end flow control [7]. This convergence is made possible by the emergence of a new breed of *Deterministic Networks*, that is being developed to support traffic that is highly sensitive to jitter, requires bounded latency in the worst case scenario, and packet loss ratio to be reduced by multiple orders of magnitude compared to existing Internet technologies. This class of network is the scope of a WG being formed at the IETF, DetNet<sup>3</sup> which aims at enabling end-to-end deterministic paths over bridges and routers.

<sup>1</sup> <http://tools.ietf.org/wg/6lo/charters>

<sup>2</sup> <http://tools.ietf.org/wg/roll/charters>

<sup>3</sup> <https://www.ietf.org/mailmain/listinfo/detnet>

In future IoT applications, deterministic WSN technologies will have to share bandwidth and other physical resources with non-deterministic traffic, for reaching higher scales at lower costs. The IETF 6TiSCH WG<sup>4</sup> addresses this additional challenge and allows for a mix of stochastic (best effort) IPv6 flows with such well-known deterministic flows while preserving their deterministic properties regardless of the overall load as imposed by other flows. 6TiSCH combines SDN-type centralized routing, for time-sensitive flows, with distributed routing and scheduling based on the Routing Protocol for Low-Power and Lossy Networks (RPL) [8], for ancillary flows.

Despite all the traditional SDN standardization activities that have been undertaken so far by different Standard Development Organizations (SDOs) and open source organizations [9], [10], DetNet and 6TiSCH are one of the first WGs addressing SDN issues (centralized routing, resource allocation) in an IoT context. With this paper we aim to present the work we are carrying within the two WGs, and the challenges we will be facing.

The rest of the paper is organized as follows. In Sec. II, we motivate the need of scheduling for making wireless IoT network deterministic. Then, in Sec. III, we present the SDN architecture proposed for this kind of networks, by DetNet. In Sec. IV, we describe the IoT 6TiSCH architecture, and its centralized scheduling approach. In Sec. V, we outline all the work that still has to be done within DetNet and 6TiSCH in order to build future SDN-based IoT deterministic networks. Finally, with Sec. VI, we conclude the paper.

## II. SCHEDULING: THE KEY TO MAKE WIRELESS DETERMINISTIC

Deterministic networking refers to the pre-computation and pre-allocation of pre-determined physical resources in the network (queues, buffers, transmission medium) for well characterized flows that are known a-priori, in order to avoid statistical effects that lead to poor bandwidth utilization, uncontrolled jitter, and congestion loss. A path is nailed down for a particular set of resources at particular times, and the forwarding behavior ensures that the right packets are forwarded at the right time to make use of these resources. When the expected packet does not arrive at the particular scheduled time, the resources allocated at that time are freed and may be used for best effort traffic.

A foremost goal for deterministic networks is to *eliminate congestion loss* by maintaining at all time the amount of critical packets within the physical capabilities of the devices. This can be achieved by the use of time-shared resources (bandwidth and buffers) per circuit, and/or by shaping and/or scheduling network activities at every hop. The associated goal is to *guarantee a worst case latency* whatever network conditions. This is achieved through fully time-controlled operations whereby the position of a given packet is known at all times and thus the amount of packets in a device buffers and queues is also known at all times. Finally, deterministic networks may

also (nearly) *eliminate equipment failure losses* using packet elimination, retry (over multi-path) and replication techniques.

A real-world network cannot be mathematically deterministic since there is no way to guarantee a perfect delivery ratio, and radios are worse than wires due to high rates of losses in transmission. What can be made deterministic is the bounded worst case latency, and the delivery ratio can only be optimized.

Beyond the advantages offered to traditional OT wired networks, a deterministic approach can provide additional values to wireless networks. Scheduling reduces transmission losses (by exploiting time and frequency diversity), optimizes the bandwidth usage, and enables a better energy conservation: by synchronizing sender and listener, it is possible to maintain peer devices in deep sleep between scheduled transmissions; therefore, the waste of power spent in idle listening and long preambles is eliminated.

In wireless, a technique like the IEEE802.1Qbu, Frame Preemption [11] is not really workable, because the talker on the shared medium might not be self, and there is usually no way to interrupt a remote talker that is blinded by its own signal. Moreover, depending on the specific radio technology, multiple packets may be burst for a single transmission opportunity, and data rates may vary dramatically depending on distance and environment; the exact duration of a transmission may thus be very hard to predict and expensive to guard against by emptying the medium prior to the transmission for that duration.

The net is that neither preemption nor guard time is really applicable to wireless, and the only way to make wireless deterministic is to *schedule* all the transmissions. The most advanced Industrial WSNs schedule at which time and on which channel a frame is forwarded by which hop. With fully scheduled operations, it is thus possible to guarantee deterministically the time of delivery.

Scheduling implies that all nodes control the precise time of emission, which in turn requires a shared and precise sense of time. On the ISM band, it is not possible, though, to guarantee that all possible interferer will be scheduled. There will be all sorts of co-channel interference, and the worst of all is often the self-inflicted multi-path fading, which is due to reflections of the transmission itself. So there is no way to guarantee the delivery of all frames.

To improve packet delivery ratio, different mitigation approaches should be adopted. The key to improve the delivery ratio and mitigate all sorts of unpredictable interferences is *diversity*. All possible forms of diversity should be leveraged, in the *spatial* domain by routing over multi-path or using parallel transmissions (i.e., Packet Replication Techniques, PRT), in the *temporal* domain by retrying failed transmissions (i.e., Automatic Repeat Request, ARQ), and in the *frequency* domain with channel hopping.

## III. THE DETNET SDN MODEL

Knowing that scheduling is the key for making wireless networks deterministic, there is now need to define a network

<sup>4</sup> <http://tools.ietf.org/wg/6tisch/charters>

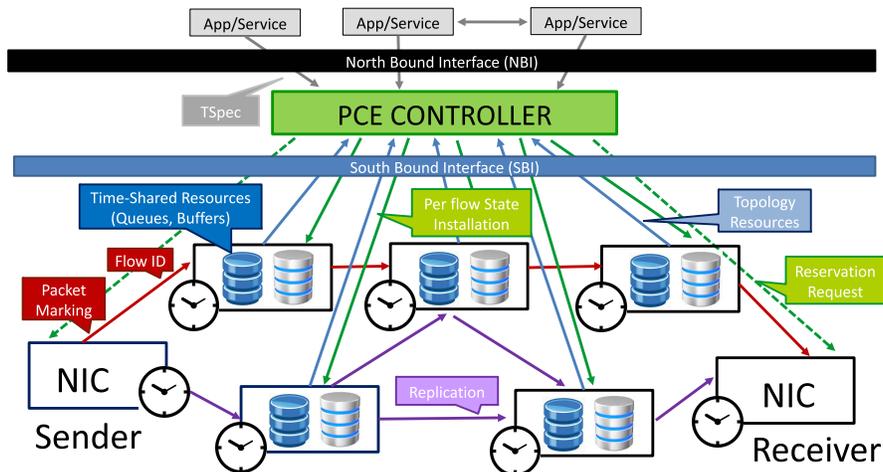


Fig. 1. DetNet SDN model for deterministic networks. From a controller perspective, the network of interconnected routers and switches (i.e., SDN data plane) looks like a collection of boxes joined by links, both of which exhibit a certain suite of physical characteristics and capabilities, like buffers and timers (clocks) for boxes and bandwidth and signal quality for links.

model which enables a fully scheduled operation orchestrated by a central controller, while supporting a more distributed operation with probably lesser capabilities.

DetNet, a new WG being formed at IETF, initially considers an SDN model [12], which typically includes a PCE for the route computation functionality (Fig. 1). The choice of such model leverages on the fact that SDN, as defined in [13], can facilitate the design, delivery, and operation of network services in a *deterministic, dynamic, and scalable* manner. Therefore, it is the *right* approach for designing future IoT deterministic networks.

The DetNet group will define an overall SDN architecture, and will focus on the signaling elements to be used to establish a deterministic path in the network from a sender to a receiver, by a centralized controller, as shown in Fig. 1. This includes: (i) the definition of data models to report the topology and the devices capabilities to the controller; (ii) the protocol elements to request a path set up for a given flow and configure the Network Interface Card (NIC) in the end nodes; and (iii) the time-shared reservation of physical resources in the network nodes along the end-to-end path. Moreover, it includes the data models to set up a path that supports packet replication, retry and elimination mechanisms to ensure high reliability through spatial and temporal diversity, and the definition of tagging elements (i.e., Flow ID, and packet marking) to be used to identify the flow that is to be forwarded along that path. New protocols will be defined, or/and other already existing relevant technologies, such as PCE and MPLS will be extended, when suitable.

#### IV. 6TiSCH DETERMINISTIC NETWORKS

In industrial WSNs that have been deployed so far, all possible transmissions are scheduled in order to maintain the medium available at critical times. This is achieved through variations of Time Division Multiplexing (TDM), which con-

sists in slicing time and assigning time slots to particular transmissions. The scheduled mode of operation is particularly adapted to well known, periodic flows, for which a schedule can be computed in advance.

The Time Slotted Channel Hopping (TSCH) technique, which was pioneered by DUST Networks with the TSMP protocol [14], is used in WirelessHART and ISA100.11a. TSCH combines TDM with a form of frequency agility called Channel Hopping in order to defeat all forms of interferences. TSCH is particularly efficient against multi-path fading, which generally affects 2 or 3 out of the 16 channels available with IEEE802.15.4 in the 2.4GHz band [15]. In this case, the schedule indicates not only the time of individual transmissions, but also the channels where the transmissions take place; a given transmission occurs on a given channel, but an eventual retransmission will occur on a different channel that is selected with a pseudo-random algorithm.

The 6TiSCH architecture, defined for an IPv6 Multi-Link subnet, shown in Fig. 2, is composed of a high speed powered backbone and a set of IEEE802.15.4 TSCH low-power wireless networks (LLNs) attached and synchronized by Backbone Routers (6BBRs) [16]. Time in a TSCH network is spliced up into time slots, which are grouped into one or several slotframes which continuously repeat over time.

The 6TiSCH WG at the IETF has proposed an IPv6-enabled architecture for Industrial IoT applications, and aims to supports best effort traffic on deterministic TSCH-based networks. The 6TiSCH Architecture [16] defines a remote monitoring and scheduling management of a TSCH network by a PCE, which cooperates with an abstract Network Management Entity (NME) to manage time slots and device resources in such a way that the load (and thus, energy consumption) placed on the constrained devices is minimized. The PCE can lock some resources (namely hard cells) for deterministic flows along paths called tracks, so that traffic on a track cannot be

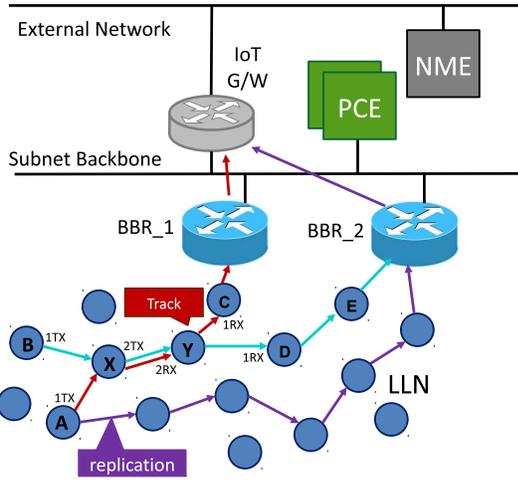


Fig. 2. Example of a 6TiSCH architecture and tracks established between field nodes and an IoT gateway.

influenced whatever by other (controlled) flows. The remaining soft cells (i.e., time/frequency resources) are made available for best effort traffic.

#### A. 6TiSCH Schedule, Bundles and Tracks

All nodes in the TSCH network are synchronized to a slotframe, and communicate by following a common schedule, which looks like a matrix of cells, each of them identified by a slot offset and a channel offset (Fig. 3). The schedule specifies at what time and on which frequency two neighbors can exchange data. Thus, a cell can be un-scheduled (no communication will take place in that cell), or scheduled, either in transmission (TX) or in reception (RX). If there is a lot of data flowing between two neighbors nodes, the schedule may contain multiple cells, assigned to them.

6TiSCH defines the peer-wise concept of a *bundle*, needed for the communication between adjacent nodes. A bundle is a group of equivalent scheduled cells, i.e., cells identified by different [slot Offset, channel Offset], which are scheduled for a same purpose, with the same neighbor, with the same flags, and the same slotframe. The size of the bundle refers to the number of cells it contains. Given the length of the slotframe, the size of the bundle translates directly into bandwidth. Finally, a bundle represent a half-duplex link between nodes, one transmitter and one or more receivers, with a bandwidth that amount to the sum of the time slots in the bundle. Bundles, thus, come in pairs, an incoming and an outgoing.

If the Backbone is Deterministic, then the 6BBR has to ensure that the end-to-end deterministic behavior is maintained between the LLN and the backbone. To this aim, the PCE should be able to compute a deterministic path end-to-end across the TSCH network and the backbone, while DetNet protocols elements should enable to program the deterministic forwarding operations in the devices.

6TiSCH has defined the concept of a *track* as a unidirectional path between a source and a destination. In the

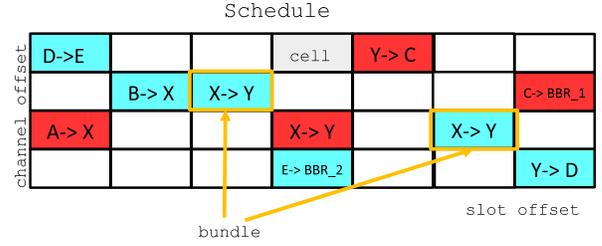


Fig. 3. Example of TSCH schedule

example in Fig. 2, a track is established from a field device in a 6TiSCH network to an IoT gateway that is located on a deterministic backbone. At each 6TiSCH hop along the track, the PCE may schedule a bundle, in order to support L2 retries (ARQ). To forward, and thus, delivery a packet to a next hop (or to an upper layer in the node), the 6TiSCH architecture supports three different forwarding models. Among these, the Track Forwarding (TF) is the one adopted in a centralized architecture, under the control of a PCE.

Along a track, a bundle of cells set to receive (RX-cells) is uniquely paired to a bundle of cells that are set to transmit (TX-cells), representing a L2 forwarding state that can be used regardless of the network layer protocol. This model can effectively be seen as a Generalized Multi-protocol Label Switching (G-MPLS) operation in that the information used to switch a frame is not an explicit label, but rather related to other properties of the way the packet was received, a particular cell in the case of 6TiSCH. A track is thus formed end-to-end as a succession of paired bundles, a RX bundle from the previous hop and a TX bundle to the next hop along the track, and a cell in such a bundle belongs to at most one track (see Fig. 2). A bundle is globally identified by (source MAC address, destination MAC address, track ID). For each segment, along a given track, for instance A-X-Y, there are two bundles in node X, one incoming (macA, macX, track ID) and one outgoing (macX, macY, track ID). Finally, pair of bundles at L3 forms a link. For instance, by using the constant NULLT as track ID for a L3 link, an IP link between two adjacent nodes X and Y, will comprise two bundles: (macX, macY, NULLT), and (macY, macX, NULLT).

The bundles may be computed so as to accommodate both variable rates and retransmissions, so they might not be fully used at a given iteration of the schedule. The 6TiSCH architecture provides additional means to avoid waste of cells as well as overflows in the transmit bundle. On one hand, a TX-cell that is not needed for the current iteration may be reused opportunistically on a per-hop basis for routed packets. When all the frame that were received for a given track are effectively transmitted, any available TX-cell for that track can be reused for upper layer traffic for which the next-hop router matches the next hop along the track. On the other hand, it might happen that there are not enough TX-cells in the X bundle to accommodate the track traffic, for instance if more

retransmissions are needed than provisioned. In that case, the frame can be placed for transmission in the bundle that is used for L3 traffic towards the next hop along the track as long as it can be routed by the upper layer, that is, typically, if the frame transports an IPv6 packet. It results that a frame that is received over a L3 bundle may be in fact associated to a track. In a classical IP link such as an Ethernet, off-track traffic is typically in excess over reservation to be routed along the non-reserved path based on its QoS setting. But with 6TiSCH, since the use of the L3 bundle may be due to transmission failures, it makes sense for the receiver to recognize a frame that should be re-tracked, and to place it back on the appropriate bundle if possible.

## V. 6TiSCH CHALLENGES FOR DETNET

Industrial and Automation Systems rely on a separate Human/Machine Interface (HMI) to set up their Operation Control Systems. The control loops are defined by their endpoints, based on their requirements, such as e2e latency and reliability. The latter are then turned by a centralized controller into networking characteristics that the network devices can understand, such as flow identification, time of arrival, burst size, period, and whether flow replication and elimination should be operated. Once an optimized overall schedule is computed, the controller provisions the network to establish the forwarding state in each individual devices. In this case, the path computation and the protocol that establishes the flows are centralized. In the case of 6TiSCH - and for DetNet at large -, the path computation is expected to be centralized; a PCE will obtain the topology and the device capabilities from the network, and turn that information into a TSCH schedule for each node.

Some topologies are mostly stable, and a full schedule may be computed before the real operations even start. In that case, a device may be provisioned directly by the PCE with the aggregated schedule that incorporates all the flows that will traverse that node, indicating the set of TX timeslots, RX timeslots, how to bridge them together, including rules for replication and elimination. Work at 6TiSCH [17] describes how that operation will be done over the Constrained Application Protocol (CoAP) [18] using a simplified NetConf protocol, called CoAP Management Interfaces (CoMI) [19].

But in the case where a new path must be established dynamically during the network operational runtime, an hybrid mode that combines centralized and distributed path setup will be needed. Individual tracks may then be added, modified or removed, on-demand, using a distributed protocol.

While 6TiSCH is responsible for the lower layer protocol between neighbors to negotiate communication cells, the overall work of defining the generic data models to represent the topologies and device capabilities, and the protocols to install the relevant state in the networking devices, is more generic, and will be done at DetNet before it is instantiated for 6TiSCH. Towards that goal, 6TiSCH exposes its requirements in [20].

### A. Packet Marking and Handling

The 6TiSCH Architecture describe how the packet tagging and marking is expected to happen in 6TiSCH networks. In detail, for packets that are routed by a PCE along a track, the tuple formed by the IPv6 source address and a local RPLInstanceID is tagged in the packets to identify uniquely the track and associated transmit bundle of cells. Thus, it results that the tagging that is used for a DetNet flow outside the 6TiSCH LLN should be swapped into 6TiSCH formats and back as the packet enters and then leaves the 6TiSCH network.

### B. Replication, Retries and Elimination

In current deployments, a TSCH track does not support Packet Replication and Elimination (PRE) but is systematically designed as Non-Equal Cost Multi-Path (NECM) routes, to provide multiple forwarding solutions at each hop, not for the classical purpose of load balancing but for increased diversity in case of transmission failure. This means that a track is scheduled so as to ensure that each hop has at least two forwarding solutions, and the forwarding decision is to try the preferred one and use the other in case of L2 transmission failure, as detected by ARQ.

One of the work items of DetNet WG for 6TiSCH network will be to include PRE. The Replication function in the field device sends a copy of each same packet over two different branches, and the PCE schedules each hop of both branches so that the two copies arrive at approximately the same time at the other end of the path. Retries delay a packet, and on a radio network, the more retries fail, the less chances that the next retry will succeed. It make sense to limit the number of retries at each hop, and in case of a loss on one branch, the second copy of the packet over the other branch can still reach the gateway in due time. If the network is highly unreliable, the two branches can be interconnected at intermediate replication and elimination points. If an end node receives two copies of the same packet, then the elimination function in the lower stack ignores the extra packet and presents only one copy to upper layers.

Even though it expects elimination and replication of packets along a complex track, 6TiSCH has no position about how the sequence numbers would be tagged in the packet, and expects to follow more generic recommendations by DetNet.

### C. Topology and Capabilities

With regards to the topology, 6TiSCH relies on Layer-2 beacons and Layer-3 6LoWPAN Neighbor Discovery [21] to discover the neighborhood, but does not describe a protocol to express the neighborhood information to the routing component of the controller, the PCE. 6TiSCH maintains a data model of the neighbors in use, as well as a recent history of metrics that represent the link quality per neighbor, in particular the metrics that are employed in RPL operations [22]. It must be noted that for memory reasons, only a subset of the potential neighbors may actually be monitored. The neighborhood information is meant to be exported via CoAP to a NME, but it should be shared with the PCE to compute

the overall topology, and, if possible, uploaded only once from a 6TiSCH node for both (or all of) the NME(s) and PCE(s).

Knowing only the topology is not sufficient to compute a deterministic path; in any network, things like the amount of buffers, queues, and the precision of time, are needed to compute a workable schedule. As it turns out, additional constraints in LowPower Lossy Networks such as 6TiSCH networks add a degree of complexity to the knowledge that the PCE must form and the paths that it can enable.

Nodes in a 6TiSCH network usually have a very limited number of buffers available to store frames in transit, so a device that participates to multiple flows may not be able to store even one frame per flow, and will need to be scheduled for transmit and clean the buffer very soon after receive. Nodes are often equipped with a single radio, with half duplex properties, and a transmit action can not happen at the same time as a receive action. It is critical that the PCE obtains the exact capabilities of each device, as well as the topology that they form with the associated radio propagation characteristics. To enable the above, 6TiSCH needs to provide extensions to the generic DetNet models.

The PCE should also be able to evaluate the energy consumption of the 6TiSCH nodes in the various modes of deep sleep, wake, transmit and receive, as well as the capacity of the device energy stores and the eventual capability to renew that store with scavenging. With this, the PCE should compute a schedule that fits best, globally, the energy constraints of all the nodes, so as to enable an optimal duration within the constrained resources, and reduce the OPEX incurred in manpower and network interruptions for the sake of changing batteries.

## VI. CONCLUSION

The SDN paradigm, which is instantiated at the IETF with a PCE, has been already used over the last 15 years in Wireless Industrial Sensor Networks, and will continue to play a key role in future IoT networks, enabling a mix of deterministic behaviors with dynamic operations in scalable deployments. The 6TiSCH WG, and the formation of the DetNet WG at IETF, are clear indications of this trend. When this work concludes, it will be possible to establish a multi-hop path over the IP network, for a particular flow with precise timing and throughput requirements, and carry this particular flow along the multi-hop path with such characteristics as bounded latency and ultra-low jitter, with duplication and elimination of packets over non congruent paths for a higher delivery ratio, and/or zero congestion loss, whatever the network load, and with no influence whatsoever from any other flow in the network.

## ACKNOWLEDGEMENTS

This publication was partially supported by the CoSDN project, INTER/POLLUX/12/4434480, funded by the Fonds National de la Recherche Luxembourg.

## REFERENCES

- [1] M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. A. Grieco, G. Boggia, and M. Dohler, "Standardized protocol stack for the internet of (important) things," *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 3, pp. 1389–1406, 2013.
- [2] Z. Qin, G. Denker, C. Giannelli, P. Bellavista, and N. Venkatasubramanian, "A software defined networking architecture for the internet-of-things," in *Network Operations and Management Symposium (NOMS), 2014 IEEE*. IEEE, 2014, pp. 1–9.
- [3] N. Omnes, M. Bouillon, G. Fromentoux, and O. Grand, "A programmable and virtualized network & its infrastructure for the internet of things: How can nfv & sdn help for facing the upcoming challenges," in *Intelligence in Next Generation Networks (ICIN), 2015 18th Int. Conf.*. IEEE, 2015, pp. 64–69.
- [4] F. Granelli, A. Gebremariam, M. Usman, F. Cugini, V. Stamati, M. Al-itska, P. Chatzimisios *et al.*, "Software defined and virtualized wireless access in future wireless networks: scenarios and standards," *Communications Magazine, IEEE*, vol. 53, no. 6, pp. 26–34, 2015.
- [5] N. Feamster, J. Rexford, and E. Zegura, "The road to sdn: An intellectual history of programmable networks," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 2, pp. 87–98, Apr. 2014.
- [6] A. Farrel, J.-P. Vasseur, and J. Ash, "A path computation element (pce)-based architecture," RFC Editor, RFC 4655, August 2006.
- [7] M. R. Palattella, P. Thubert, X. Vilajosana, T. Watteyne, Q. Wang, and T. Engel, "6tisch wireless industrial networks: Determinism meets ipv6," in *Internet of Things*. Springer, 2014, pp. 111–141.
- [8] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struijck, J. Vasseur, and R. Alexander, "Rpl: Ipv6 routing protocol for low-power and lossy networks," RFC Editor, RFC 6550, March 2012.
- [9] J. M. Halpern, "Standards collisions around sdn," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 10–15, 2014.
- [10] T. Egawa, F. Schneider, M. Scholler, S. Hayano, S. Schaller, and F. Zdarsky, "Standardizations of sdn and its practical implementation," *NEC Technical Journal, Special Issue on SDN and Its Impact on Advanced ICT Systems*, vol. 8, no. 2, pp. 16–20, 2014.
- [11] IEEE std. 802.1Qbu draft std, *Standard for Local and Metropolitan Area Networks-Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks - Amendment: Frame Preemption*, IEEE standard for Information Technology, June 2015.
- [12] E. Haleplidis, K. Pentikousis, S. Denazis, J. H. Salim, D. Meyer, and O. Koufopavlou, "Software-defined networking (sdn): Layers and architecture terminology," RFC Editor, RFC 7426, January 2015.
- [13] M. Boucadair and C. Jacquenet, "Software-defined networking: A perspective from within a service provider environment," RFC Editor, RFC 7149, March 2014.
- [14] K. Pister and L. Doherty, "Tsmpt: Time synchronized mesh protocol," in *IASTED Int. Symposium on Distributed Sensor Networks (DSN08)*. Orlando, Florida, USA: IEEE, Nov. 2008.
- [15] IEEE std. 802.15.4, Part. 15.4: *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE standard for Information Technology, Sep. 2009.
- [16] P. Thubert, "An architecture for ipv6 over the tsch mode of ieee 802.15.4," Working Draft, IETF Secretariat, Internet-Draft draft-ietf-6tisch-architecture-08, May 2015.
- [17] R. Sudhaakar and P. Zand, "6tisch resource management and interaction using coap," Working Draft, IETF Secretariat, Internet-Draft draft-ietf-6tisch-coap-03, March 2015.
- [18] Z. Shelby, K. Hartke, and C. Bormann, "The constrained application protocol (coap)," RFC Editor, RFC 7252, June 2014.
- [19] P. V. der Stok, A. Bierman, J. Schnwider, and A. Sehgal, "Coap management interface," Working Draft, IETF Secretariat, Internet-Draft draft-vanderstok-core-comi-07, July 2015.
- [20] P. Thubert, "6tisch requirements for detnet," Working Draft, IETF Secretariat, Internet-Draft draft-thubert-6tisch-4detnet-01, June 2015.
- [21] Z. Shelby, S. Chakrabarti, E. Nordmark, and C. Bormann, "Neighbor discovery optimization for ipv6 over low-power wireless personal area networks (6lowpans)," RFC Editor, RFC 6775, November 2012.
- [22] J. Vasseur, M. Kim, K. Pister, N. Dejean, and D. Barthel, "Routing metrics used for path calculation in low-power and lossy networks," RFC Editor, RFC 6551, March 2012.