IETF 6TSCH: Combining IPv6 Connectivity with Industrial Performance

Pascal Thubert*, Thomas Watteyne†, Maria Rita Palattella‡, Xavier Vilajosana§ and Qin Wang¶
* *Cisco Systems, France, Email: pthubert@cisco.com*
† *Linear Technology/Dust Networks, CA, USA, Email: twatteyne@linear.com*
‡ *SnT, University of Luxembourg, Luxembourg, Email: maria-rita.palattella@uni.lux*
§ *Universitat Oberta de Catalunya, Catalonia, Spain, Email: xvilajosana@uoc.edu*
¶ *University of Science and Technology Beijing, China, Email: wangqin@ies.ustb.edu.cn*

***Abstract*—This paper presents a new work called 6TSCH that is starting at the IETF to enable a large IPv6 multi-link subnet with industrial-grade performances of jitter, latency and reliability. The subnet is composed of a high speed powered backbone and a number of IEEE802.15.4e TSCH wireless mesh networks attached and synchronized by specialized Backbone Routers. Route Computation may be achieved in a centralized or in a distributed fashion, and tracks are installed to forward well-known flows with deterministic properties along their multi-hop path.**

***Keywords*-6TSCH, IEEE802.15.4e TSCH, RPL, 6LoWPAN.**

I. INTRODUCTION

Low power and Lossy Networks (LLNs) allow to interconnect a large number of resource-constrained devices, forming a wireless mesh network. To be connected to the Internet, a small number of border routers (BRs) usually serve as gateways between each LLN and the Internet. Such LLNs have a wide range of applications, including building and home automation, industrial process control and smart urban environments.

A chief component of these networks is the wireless communication technology. The 6LoWPAN, ROLL and CoRE IETF Working Groups have defined protocols at various layers of the LLN protocol stack, including an IPv6 adaptation layer, 6LoWPAN [1], a routing protocol, RPL [2] and a web transfer protocol, CoAP [3]. This protocol stack so far has been used with IEEE802.15.4 low-power radios [4], whose limitation in mesh-networking conditions has become apparent only recently.

To overtake such limitation, the IEEE802.15.4e standard [5] has been published in 2012 as an amendment to the IEEE802.15.4-2011 Medium Access Control (MAC) protocol [4]. Three different operative modes have been defined in the IEEE802.15.4e standard. Among them, the *Timeslotted Channel Hopping* (TSCH) mode is the latest generation of ultra-lower power and reliable networking solutions for LLNs. At its core is a medium access technique which uses time synchronization to achieve ultra low-power operation and channel hopping to enable high reliability.

Its core technology is similar to the one used in industrial networking technologies such as WirelessHART [1] or ISA100.11a [2], resulting in comparable performance. However, unlike these industrial protocols, IEEE802.15.4e TSCH focuses on the MAC layer only. This clean layering allows for TSCH to fit under an IPv6 enabled protocol stack for LLNs, like the one shown in Fig. 1. Note that TSCH does not amend the physical layer, i.e., it can operate on any IEEE802.15.4-compliant hardware.

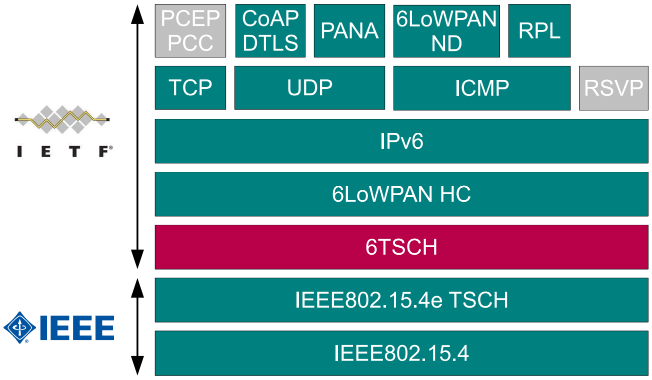


Figure 1. 6TSCH IPv6-enabled protocol stack for LLNs.

The Time-Slotted aspect of the TSCH technology is a Time Division Multiplexing (TDM) technique, which requires all nodes in the network to be time synchronized. Time is sliced up into time slots; a time slot is long enough for a MAC frame of maximum size to be sent from mote B to mote A, and for mote A to reply with an acknowledgment (ACK) frame indicating successful reception (see Fig. 2).

TSCH is different for traditional low-power MAC protocols because of its scheduled nature. All nodes in the network follow a common communication schedule which indicates for each active (transmit or receive) timeslot a channelOffset [3] and the address of the neighbor to communicate with. The channelOffset is translated into a frequency using a specific translation function which causes pairs of neighbors to "hop" between the different available frequencies when communicating. Such channel hopping

---

[1] www.hartcomm.org.

[2] http://www.isa.org/Community/ SP100WirelessSystemsforAutomation.

[3] There are as many channelOffset values as there are frequencies available, e.g. 16 when using IEEE802.15.4-compliant radios at 2.4GHz, when all channels are used.

technique efficiently combats multi-path fading and external interference.

IEEE802.15.4e only defines the link-layer mechanisms [6]. It does not define how the network communication schedule is built and matched to the traffic requirement of the network. The definition of an adaptation layer and architectural recommendations are needed for those highly efficient LLN networks to transition to end-to-end IPv6-based solutions.

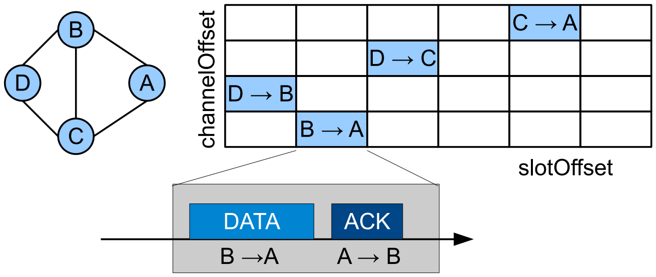


Figure 2. Example of a TSCH schedule.

To fill this gap, a new Working Group (WG) called 6TSCH [4] is being formed within the IETF. It aims to link IEEE802.15.4e TSCH capabilities with prior IETF 6LoWPAN and ROLL standardization efforts and recommendations [7]. In detail, it will (i) define an open standard-based architecture, re-using existing protocols when possible, and (ii) face networking and routing issues, among many others.

The remainder of this paper is organized as follows. Section II describes the bases for the architecture that 6TSCH WG is going to adopt. Section III introduces the concept of deterministic networks, and describes how to optimize the resource allocation in such networks. Section IV presents 6TUS, the adaptation layer proposed by 6TSCH, and its main functionalities. Section V describes the routing layer issues, using the Neighbor Discovery IPv6 (ND) [8] and the RPL protocols. Section VI concludes this paper.

II. 6TSCH ARCHITECTURE

The newly defined IETF 6TSCH group will document an open standards-based architecture, highlight best practices, and standardize the missing components to achieve industrial-grade performance in terms of jitter, latency, scalability, reliability and low-power operation for IPv6 over IEEE802.15.4e TSCH [5]. Although not addressed directly by 6TSCH, it is envisioned that the resulting techniques will be applicable to technologies other than 2.4GHz IEEE802.15.4 [4].

As illustrated in Fig. 3, the scope of the architecture is an IPv6 multi-link subnet that is spread over a high speed powered backbone and a number of IEEE802.15.4e TSCH wireless mesh networks connected to the backbone by synchronized backbone routers (BBRs).

The 6TSCH architecture [9] will specify how packets that belong to a deterministic IPv6 flow are marked and routed or forwarded over the mesh within jitter and latency budgets. It will also cover security, link management for the IPv6 network layer, neighbor discovery and routing.

As detailed in Fig. 1, when possible, the 6TSCH group will reuse existing protocols such as IP6 Neighbor Discovery (ND) [8], IPv6 Low power Wireless Personal Area Networks (6LoWPAN) [1], and the Routing Protocol for Low Power and Lossy Networks (RPL) [2], with the minimum adaptation required to meet criteria for reliability and determinism within the mesh, and scalability over the backbone.

Route Computation will be achieved either in a centralized fashion by a Path Computation Entity (PCE), which is located either on the backbone or farther in the IPv6 network over a backhaul, or in a distributed fashion using RPL and a multi-path resource reservation protocol.

Moreover, *Tracks*[5] *Allocation* can be globally optimized and then pushed on the network from the PCE that computes the routes, and/or managed by a distributed scheduling protocol along routes that are computed by RPL.

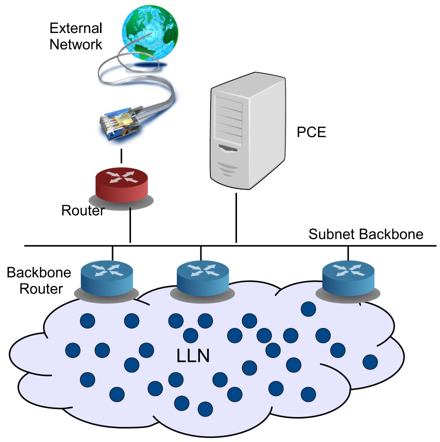


Figure 3. 6TSCH reference architecture.

The architecture will specify a framework for scheduling frames over time slots that supports three models:

1) a centralized route computation that builds and maintains the communication schedule, and distributes it to the nodes. This schedule includes forwarding information associated to time slots; RPL operations only apply to emergency repair actions when the reference topology becomes unusable. A number of existing protocols could be extended to push the schedule from the PCE to the device, including the PCE Communication Protocol (PCEP) [11], Forwarding and Control

---

[4] Online at http://www.ietf.org/mailman/listinfo/6tsch.

[5] According to 6TSCH terminology [10], a track is a deterministic sequence of cells, along a multi-hop path.

Element Separation (ForCES) [12], Software-Defined Networking (SDN) OpenFlow [6] or even through network management over the Constrained Application Protocol (CoAP) [3].

2) a distributed resource reservation and signaling protocol that establishes tracks between source and destination nodes along multi-hop routes identified by RPL. The track may be setup by extensions to the legacy Resource ReSerVation Protocol (RSVP) [13] or the more recent but rather heavy Next Steps in Signaling (NSIS) protocol [14].
3) a best effort resource allocation that is used to transport data frames on a per hop basis in the absence of a reservation protocol.

The architecture will also address how multiple BBRs are supported for a higher degree of scalability and reliability, and how nodes maintain synchronization in the presence of multiple BBR. This work implies new IPv6 ND operations as detailed in Sec.V.

As a result, the 6TSCH architecture will enable a new range of use cases for LLNs, including: (a) Control loops in a wireless process control network, in which high reliability and a fully deterministic behavior are required; (b) Umbrella urban networks transporting data from different independent clients, and for which an operator needs flow isolation and traffic shaping; (c) Energy harvesting networks, which require an extremely low and predictable average power consumption.

III. DETERMINISTIC BEHAVIOR AND OPTIMIZATIONS

Determinism applies to traffic flows with an emission rate and routing path patterns that are well-known in advance. For such traffic, a *deterministic network* allocates the required resources (buffers, processors, medium access) along the multi-hop routing path at the precise moment the resources are needed. The forwarding elements can thus handle any given frame or packet with a jitter that is negligible with regards to the particular application.

A good example of a deterministic network is a railway system. A railway system is deterministic because trains are scheduled periodically to leave a railway station at a certain time, to traverse the stations along a predetermined track at very precise times as well so that, in fine, a given train arrives at its final station at the exact expected time, with virtually no jitter from a human perspective. Collision are eliminated and there is never another train blocking the rail and delaying this train.

Industrial Process Control frequently uses 1Hz to 4Hz control loops, and for those, the MAC protocol can be considered deterministic, even when clocks drift in the order of tens of ppm. A low-throughput technology such as IEEE802.15.4 is thus well adapted; the bandwidth can be pre-formatted in a Time Division Multiplexing (TDM) fashion, and time slots become a unit of throughput that can allocated to a deterministic flow, without incurring a huge consumption of system resources.

On the other hand, Factory Automation can be a hundred times faster, and is often addressed with a deterministic variation of the Ethernet protocol. For such a fast medium, the precise moments when the forwarding entity really needs to care about deterministic packets become very sparse along the bandwidth, and formatting it all in time slots with enough resolution for the acceptable jitter would incur undue costs. Rather, Deterministic Ethernet places landmarks at the specific points in the future where a deterministic event is scheduled to occur, which requires per-flow timers.

Routing in a Deterministic Network can be operated either in a centralized or in a distributed fashion, but only the centralized routing operation can guarantee the overall optimization for all the flows with a given set of constraints and goals. The centralized computation can be done by a Controlling Element in a SDN or ForCES architecture. In a Multi-protocol Lambda Switching architecture [15], it is typically done by a PCE with an Objective Function that represents the goals and constraints. It is already typical for a PCE to compute not only an optimized Layer 3 path for traffic Engineering, but also to provide the actual Lambda layout for the lower layers. In a similar fashion, it would make sense to extend the PCE to compute time slots associated with a deterministic flow at the same time as it computes a route over the LLN. This requires a knowledge of the flows as well as a knowledge of the radio behavior at each hop; for instance, an estimation of the expected transmission count (ETX) so as to provision enough time slots for retransmissions.

The design of RPL also includes the capability to build routing topologies ("instances" in RPL parlance) that are associated to Objective Functions, but in a distributed fashion. With RPL, it is still possible to impose Deterministic behavior along a routing path, with in particular an ultra-low jitter, but it is not possible to guarantee that an individual path is fully optimized, or that the distribution of resources is globally optimized. On the other hand, the routing operations will be more efficient (no need of CPU intensive PCE computations) and resilient (no dependence on a PCE for base routing and recovery).

IV. 6TUS

In the 6TSCH architecture [9], the 6tus layer [16] sits on top of IEEE802.15.4e TSCH, and allows a scheduling entity to drive the TSCH schedule. 6tus also includes statistics collection functionality, which an upper layer (including the RPL routing protocol) can use to gather connectivity information. Finally, 6tus includes a monitoring process which can flag when a particular cell, i.e., a single element in the TSCH schedule, does not perform as well as expected.

---

[6] http://OpenFlowSwitch.org.

6tus is designed to be used with several scheduling approaches. In a centralized approach, a central PCE collects topology and traffic requirements, used to build a communication schedule, which it then sends to the different nodes in the network. In a decentralized approach, nodes compute their own schedule according to local information or by using a decentralized resource reservation protocol. To enable both approaches, 6tus adds a new IEEE802.15.4e LinkOption flag [5]; in addition to the Tx, Rx, Shared and Timekeeping flag, a cell is also qualified as either a *hard cell* or *soft cell*. This option is mandatory; all cells are either hard or soft.

A *hard cell* is a cell that cannot be dynamically reallocated by 6tus. This type of cell is typically scheduled by a PCE. Once installed, only the PCE can move it inside the TSCH schedule, or delete it. When installing a hard cell, the PCE indicates the exact slotOffset and channelOffset of the cell. A *soft cell* is a cell that can be reallocated by 6tus dynamically. This type of cell is typically scheduled by a distributed scheduling entity. Instead of specifying the exact slotOffset and channelOffset, the scheduling entity indicates how many cells to schedule to a given neighbors. The monitoring process of 6tus keeps track of the performance of each of the cells to the same neighbor. If a cell performers significantly worse that the others scheduled to the same neighbor, 6tus reallocates this cells at different timeOffset and channelOffset inside the TSCH schedule.

When using a centralized scheduler, the PCE needs a protocol to send schedule updates to the nodes in the network. Candidate protocols include PCEP [11], OpenFlow, and ForCES [12]. When using a distributed scheduler, a protocol is needed to reserve MAC-level resources along the multi-hop path identified by RPL, to satisfy a certain QoS constraints (e.g. bandwidth, latency). Candidate protocols include RSVP [13], [17] or NSIS [14]. NSIS provides the semantics for transport layer packets to visit each node along a multi-hop RPL path, and indicating Quality Of Service (QoS) requirements. Upon reception of a QoS request, the 6tus layer configures the appropriate MAC layer resources.

6tus maintains statistics about the performance of scheduled cells. When using a centralized scheduler, this information is periodically sent to the PCE, which continuously adapts the schedule and sends schedules updates as needed. This information can also be used by the RPL protocol's objective function.

A 6TSCH network can transport different types of traffic, possibly for different administrative entities (e.g. lighting and HVAC data in a smart building), possibly with different QoS constraints. Thanks to the slotted nature of IEEE802.15.4e TSCH, 6tus can mark different cells with indentifiers of those different flow. This can result in perfect isolation, in which for example the amount of HVAC traffic has no effect on the latency of the lighting traffic. This allows for true "umbrella" networks, managed by a network operator, and transporting data for different clients. An example is an urban network which is used to transport data data from weather sensors, and actuation commands for the municipal sprinkler system.

When a packet enters the 6TSCH network, the 6tus layer at the ingress point identifies the service this packet belong to and marks the packet, possibly by using DSCP field in the 6LoWPAN header. When traveling through the 6TSCH network, each mote will use that marker to decide on which cell to transmit.

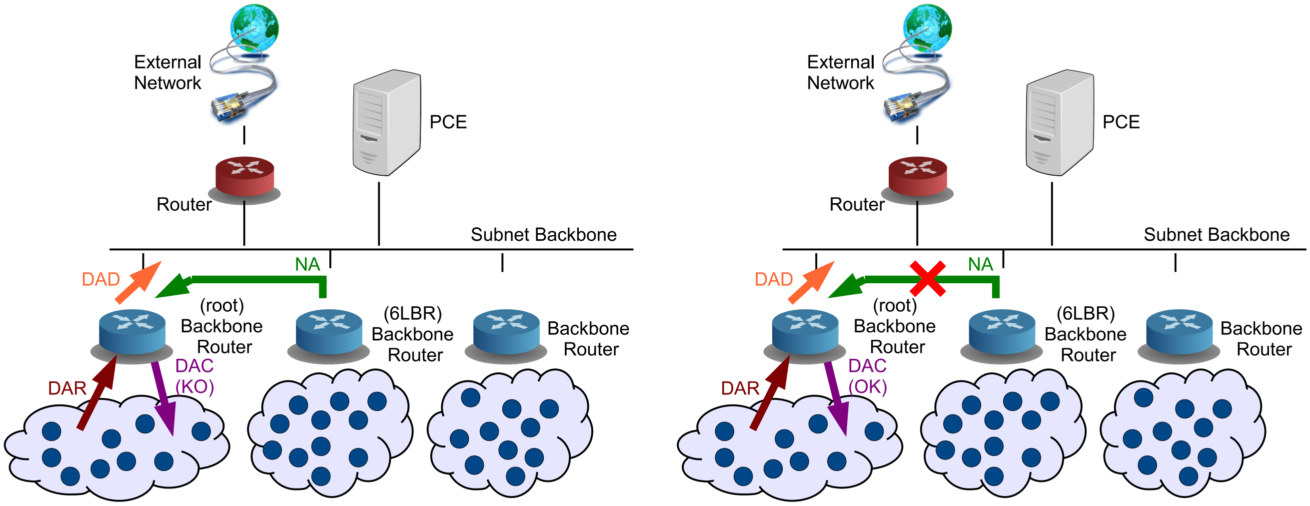
V. WIRELESS NEIGHBOR DISCOVERY AND RPL

IPv6 Neighbor Discovery (IPv6 ND) [8] operations were defined at a time when an Ethernet network was mostly a single wire, and the cost of a broadcast was roughly the same as that of a unicast, in that it would use the whole wire anyway. For that reason, IPv6 ND activities are largely based on multicast, which is not adapted to the large switched infrastructures that we see today in corporate and data center environments, and certainly not adapted to wireless clients that move permanently. Each movement involves one or more multicast messages for IPv6 ND operations alone, which clogs the wireless medium all around the switched fabric at a speed that is much lower than that used for unicast, thus wasting huge amounts of unicast-equivalent bandwidth.

Neighbor Discovery Optimization for Low-power and Lossy Networks [18] addresses this issue with a new model of IPv6 ND registration. This model is generalized to devices connected at layer 2 to the switched fabric in the Efficiency-Aware IPv6 Neighbor Discovery Optimizations [19]. The key is that multicast flooding, traditionally required for Duplicate Address Detection (DAD), is replaced by a unicast registration to a centralized binding table. The new message to create an entry is a Duplicate Address Request (DAR) that is answered by a Duplicate Address Confirmation (DAC) message. Finally, a new Address Registration Option (ARO) is introduced which contains a unique ID for the device, typically an EUI-64 address.

The IPv6 ND registration mechanism is a fundamental change of paradigm, and a quantum leap forward to address wireless devices. In particular, the ND registration is simpler than DHCP, since it is still an auto-configuration model; and yet it may provide benefits that fall traditionally in DHCP-land in terms of control by network operators or some automation. With ND registration, there is still an entity (called the LLN Border Router, LBR) that grants the right to use an address for a lifetime, and it is possible to lookup from an administrative perspective who owns which address in a much more deterministic fashion than the classical snooping done for Source Address Validation Improvements (SAVI).

At the same time, the registration mechanism needs to be extended to address the actual needs of large deployments



(a) Duplication: different EUI-64 in ARO option.

(b) Mobility: same EUI-64 (newer TID wins).

Figure 4. Determination of duplication vs. mobility.

such as an extensive factory floor. In such a case, renumbering is not an option and an IPv6 subnet may grow to the thousands. A classical model to scale such a network consists in laying out a high-speed backbone that spans the area, with backbone routers (BBRs) [20] placed along the backbone. Each BBR becomes the root for the RPL-based mesh of devices around it.

In such a network, it makes sense to distribute (and inter-work) the LBR registry functionality, for instance by collocating the LBR for wireless devices attached to a RPL DODAG with the BBR that acts as root for that DODAG.

An issue arises when the same address is registered asynchronously on two different BBRs, as it is unclear whether that is a device that just moved, or if it is an address that is duplicated between two devices. Whether the BBR inter-working is done through a routing protocol or classical IPv6 ND, there is a need for an extension to assert this. RPL, and to that regard any traditional routing protocol, will not consider that two advertisements can represent a duplication, but simply that there are probably two ways to get to the same device. On the other hand, the ARO in [18] can be used to find out when there is a duplication though a device unique ID as illustrated in Fig. 4(a), but cannot tell which is the current state that must be conserved from the stale one, that should be cleaned up. RPL uses a sequence counter (called “DAOSequence”) to detect stale advertisements, and there is probably a need to enhance the ARO to add a similar indication (a Transaction ID, TID) for use within the ND registration mechanism, as illustrated in Fig. 4(b).

Once the duplication problem is sorted out, there is still a need to discover and route over the backbone between a device that is attached to the backbone and a wireless device that is located inside a DODAG and reachable over the BBR. It is possible to extend RPL over the backbone and present the subnet as not-onlink in Router Advertisements, so as to always route, over the backbone and then along the DODAG. The alternate is a mixed mode over the backbone that consists in proxying ND operations. Upon a RPL route advertisement (called a “DAO”), the BBR that acts as root for the DODAG where a given device is located installs a host route towards the device over the LLN. Then, it advertises the device’s address over the backbone using classical ND with extensions to check for duplication and movement. In this way, any legacy IPv6 device, using the classical IPv6 ND exchange of a Neighbor Solicitation (NS) and its Neighbor Advertisement (NA) response, resolves that the MAC address for the device is in fact that of the BBR. Then, it passes on the packet, which the BBR finally routes over the DODAG to the wireless destination. This procedure is illustrated in Fig. 5.

There are a number of questions to be answered in this proxy ND operation. In particular, how does this model work with multiple instances that are eventually rooted at different BBRs, and there are well-known possible answers such as the use of VLANs. Regardless of the answer, there is substantial work to be done to extend the simple model in [18] to operate over a backbone, and then enable routing from the backbone towards a LLN device. The term “WiND” – for “wireless ND” – was coined to refer to this situation, and which certainly applies in part to 6TSCH.

VI. CONCLUSION

Existing industrial Wireless Sensor Network technologies have demonstrated that the IEEE802.15.4e *Timeslotted Channel Hopping* (TSCH) effectively enables industrial-grade deterministic properties for slow speed control loops with low latency, ultra-low jitter and a high reliability. It makes sense to extend this support, that is essentially based

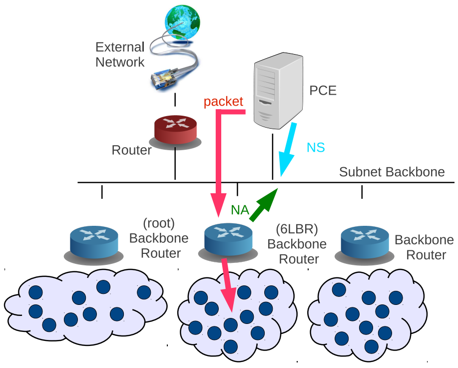


Figure 5. ND Resolution.

on centralized routing, to a distributed mode that can be cheaper, at the expense of the optimization that only a centralized approach can obtain. The IETF is now starting an effort, called 6TSCH, which will provide both centralized and distributed operation, based on open standards, and which will enable a new range of applications in automation (home, city, building) and man-to-machine interfaces (cars, planes), thus optimizing processes and saving energy and resources for a greener planet.

ACKNOWLEDGMENT

This publication was supported by the FP7 project IoT6-288445, partially funded by the European Community. Xavier Vilajosana is funded by the Spanish Ministry of Education under Fullbright-BE grant (INF-2010-0319).

REFERENCES

[1] N. Kushalnagar, G. Montenegro, and C. Schumacher, IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals, RFC 4919, Internet Engineering Task Force RFC 4919, August 2007.

[2] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. P. Vasseur, and R. Alexander, RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks, RFC 6550, Internet Engineering Task Force RFC 6550, March 2012.

[3] Z. Shelby, K. Hartke, C. Bormann, and B. Frank, Constrained Application Protocol (CoAP), IETF CoRE Working Group, February 2011.

[4] IEEE802.15.4-2011: IEEE Standard for Local and metropolitan area networks – Part. 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks(LR-WPANs), Standard for Information Technology Std., September 2011.

[5] IEEE802.15.4e-2012: IEEE Standard for Local and Metropolitan Area Networks. Part 15.4: Low-Rate Wireless Personal Area Networks (LRWPANs) Amendment 1: MAC Sublayer, Institute of Electrical and Electronics Engineers Std., 16 April 2012.

[6] T. Watteyne, Ed., M.R. Palattella, L.A. Grieco, Using IEEE802.15.4e TSCH in an LLN context: Overview, Problem Statement and Goals. draft-watteyne-6tsch-tsch-lln-context-01 (work in progress), February 2013.

[7] T. Phinney, P. Thubert, RA. Assimiti, RPL applicability in industrial networks. draft-phinney-roll-rpl-industrial-applicability-00, October 2011

[8] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, Neighbor Discovery for IP version 6 (IPv6), RFC 4861, September 2007.

[9] Thubert, P., Ed., Assimiti, RA., Watteyne, T., An Architecture for IPv6 over Time Synchronized Channel Hopping. draft-thubert-6tsch-architecture-00 (work in progress), March 2013.

[10] Palattella, MR., Ed., Thubert, P., Watteyne, T., and Q. Wang, Terminology in IPv6 over Time Slotted Channel Hopping. draft-palattella-6tsch-terminology-00 (work in progress), March 2013.

[11] JP. Vasseur, JL. Le Roux, Path Computation Element (PCE) Communication Protocol (PCEP) RFC 5440, Internet Engineering Task Force RFC 5440, September 2009.

[12] A. Doria, J. Hadi Salim, R. Haas, H. Khosravi, W. Wang, L. Dong, R. Gopal, J. Halpern, Forwarding and Control Element Separation (ForCES) Protocol Specification RFC 5810, Internet Engineering Task Force RFC 5810, September 2010.

[13] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, Resource ReSerVation Protocol (RSVP) : Version 1 Functional Specification. RFC 2205, Internet Engineering Task Force RFC 2205, September 1997.

[14] R. Hancock, G. Karagiannis, J. Loughney, S. Van den Bosch, Next Steps in Signaling (NSIS): Framework. RFC 4080, Internet Engineering Task Force RFC 4080, June 2005.

[15] F. Le Faucheur, L. Wu, B. Davie, S. Davari, P. Vaananen, R. Krishnan, P. Cheval, J. Heinanen, Multi-Protocol Label Switching (MPLS)Support of Differentiated Services. Internet Engineering Task Force RFC 3270, May 2002.

[16] Wang, Q., Ed., Vilajosana, X., and Watteyne, T., 6tus Adaptation Layer Specification. draft-wang-6tsch-6tus-00 (work in progress), March 2013.

[17] A. Morell , X. Vilajosana , J.L. Vicario , T. Watteyne, Label Switching over IEEE802.15.4e Networks. Transactions on Emerging Telecommunications Technologies. April 2013

[18] Shelby, Z., Chakrabarti, S., Nordmark, E., and C. Bormann, Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs), RFC 6775, November 2012.

[19] Chakrabarti, S., Nordmark, E., and M. Wasserman, Efficiency aware IPv6 Neighbor Discovery Optimizations, draft-chakrabarti-nordmark-6man-efficient-nd-01 (work in progress), November 2012.

[20] P. Thubert, 6LoWPAN Backbone Router. draft-thubert-6lowpan-backbone-router-03 (work in progress), February 2013.