Internet Engineering Task Force (IETF)
Request for Comments: 7554
Category: Informational
ISSN: 2070-1721

T. Watteyne, Ed.
Linear Technology
M. Palattella
University of Luxembourg
L. Grieco
Politecnico di Bari

May 2015

# Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement

## Abstract

This document describes the environment, problem statement, and goals for using the Time-Slotted Channel Hopping (TSCH) Medium Access Control (MAC) protocol of IEEE 802.14.4e in the context of Low-Power and Lossy Networks (LLNs). The set of goals enumerated in this document form an initial set only.

## Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7554.

Table of Contents

## 1. Introduction

IEEE 802.15.4e [IEEE.802.15.4e] was published in 2012 as an amendment to the Medium Access Control (MAC) protocol defined by the IEEE 802.15.4 standard (of 2011) [IEEE.802.15.4]. IEEE 802.15.4e will be rolled into the next revision of IEEE 802.15.4, scheduled to be published in 2015. The Time-Slotted Channel Hopping (TSCH) mode of IEEE 802.15.4e is the object of this document. The term "TSCH" refers to TSCH as used in [IEEE.802.15.4e].

This document describes the main issues arising from the adoption of the TSCH in the LLN context, following the terminology defined in [TERMS-6TISCH]. Appendix A further gives an overview of the key features of the TSCH amendment to IEEE 802.15.4e. Appendix B details features of TSCH, which might be interesting for the work of the 6TiSCH WG.

TSCH was designed to allow IEEE 802.15.4 devices to support a wide range of applications including, but not limited to, industrial ones [IEEE.802.15.4e]. At its core is a medium access technique that uses time synchronization to achieve low-power operation and channel hopping to enable high reliability. Synchronization accuracy impacts power consumption and can vary from microseconds to milliseconds depending on the solution. This is very different from the "legacy" IEEE 802.15.4 MAC protocol and is therefore better described as a "redesign". TSCH does not amend the physical layer, i.e., it can operate on any hardware that is compliant with IEEE 802.15.4.

IEEE 802.15.4e is the latest generation of ultra-lower power and reliable networking solutions for LLNs. [RFC5673] discusses industrial applications and highlights the harsh operating conditions as well as the stringent reliability, availability, and security requirements for an LLN to operate in an industrial environment. In these environments, vast deployment environments with large (metallic) equipment cause multi-path fading and interference to thwart any attempt of a single-channel solution to be reliable; the channel agility of TSCH is the key to its ultra-high reliability. Commercial networking solutions are available today in which nodes consume 10’s of microamps on average [CurrentCalculator] with end-to-end packet delivery ratios over 99.999% [Doherty07channel].

IEEE 802.15.4e has been designed for low-power constrained devices, often called "motes". Several terms are used in the IETF to refer to those devices, including "LLN nodes" [RFC7102] and "constrained nodes" [RFC7228]. In this document, we use the generic (and shorter) term "node", used as a synonym for "LLN node", "constrained node", or "mote".

Enabling the LLN protocol stack to operate in industrial environments opens up new application domains for these networks. Sensors deployed in smart cities [RFC5548] will be able to be installed for years without needing battery replacement. "Umbrella" networks will interconnect smart elements from different entities in smart buildings [RFC5867]. Peel-and-stick switches will obsolete the need for costly conduits for lighting solutions in smart homes [RFC5826].

TSCH focuses on the MAC layer only. This clean layering allows for TSCH to fit under an IPv6-enabled protocol stack for LLNs, running an IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) [RFC6282], the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [RFC6550], and the Constrained Application Protocol (CoAP) [RFC7252]. What is missing is a functional entity that is in charge of scheduling TSCH time slots for frames to be sent on. In this document, we refer to this entity as the "Logical Link Control" (LLC), bearing in mind that realizations of this entity can be of different types, including a distributed protocol or a centralized server in charge of scheduling.

While [IEEE.802.15.4e] defines the mechanisms for a TSCH node to communicate, it does not define the policies to build and maintain the communication schedule, match that schedule to the multi-hop paths maintained by RPL, adapt the resources allocated between neighbor nodes to the data traffic flows, enforce a differentiated treatment for data generated at the application layer and signaling messages needed by 6LoWPAN and RPL to discover neighbors, react to topology changes, self-configure IP addresses, or manage keying material.

In other words, TSCH is designed to allow optimizations and strong customizations, simplifying the merging of TSCH with a protocol stack based on IPv6, 6LoWPAN, and RPL.

## 2. TSCH in the LLN Context

To map the services required by the IP layer to the services provided by the link layer, an adaptation layer is used [Palattella12standardized]. In 2007, the 6LoWPAN WG started working on specifications for transmitting IPv6 packets over IEEE 802.15.4 networks [RFC4919]. A low-power Wireless Personal Area Network (WPAN) is typically composed of a large number of battery-powered devices that are deployed at locations that are unknown a priori. Nodes form a star or a mesh topology and communicate with one another at a low datarate and using short frames. The wireless nature of the links means that they are unreliable in nature. Nodes turn off their radio interface most of the time to conserve energy. Given these

features, it is clear that the adoption of IPv6 on top of a low-power WPAN is not straightforward but poses strong requirements for the optimization of this adaptation layer.

For instance, due to the IPv6 default minimum MTU size (1280 bytes), an unfragmented IPv6 packet is too large to fit in an IEEE 802.15.4 frame. Moreover, the overhead due to the 40-byte-long IPv6 header wastes the scarce bandwidth available at the PHY layer [RFC4944]. For these reasons, the 6LoWPAN WG has defined an effective adaptation layer [RFC6282]. Further issues encompass the autoconfiguration of IPv6 addresses [RFC2460] [RFC4862], the compliance with the recommendation on supporting link-layer subnet broadcast in shared networks [RFC3819], the reduction of routing and management overhead [RFC6606], the adoption of lightweight application protocols (or novel data encoding techniques), and the support for security mechanisms (confidentiality and integrity protection, device bootstrapping, key establishment, and management).

These features can run on top of TSCH. There are, however, important issues to solve, as highlighted in Section 3.

Routing issues are challenging for 6LoWPAN, given the low-power and lossy radio links, the battery-powered nodes, the multi-hop mesh topologies, and the frequent topology changes due to mobility. Successful solutions take into account the specific application requirements, along with IPv6 behavior and 6LoWPAN mechanisms [Palattella12standardized]. The ROLL WG has defined RPL in [RFC6550]. RPL can support a wide variety of link layers, including ones that are constrained, potentially lossy, or typically utilized in conjunction with host or router devices with very limited resources, as in building/home automation [RFC5867] [RFC5826], industrial environments [RFC5673], and urban applications [RFC5548]. RPL is able to quickly build up network routes, distribute routing knowledge among nodes, and adapt to a changing topology. In a typical setting, nodes are connected through multi-hop paths to a small set of root devices, which are usually responsible for data collection and coordination. For each of them, a Destination-Oriented Directed Acyclic Graph (DODAG) is created by accounting for link costs, node attributes/status information, and an Objective Function, which maps the optimization requirements of the target scenario.

The topology is set up based on a Rank metric, which encodes the distance of each node with respect to its reference root, as specified by the Objective Function. Regardless of the way it is computed, the Rank monotonically decreases along the DODAG towards the root, building a gradient. RPL encompass different kinds of traffic and signaling information. Multipoint-to-Point (MP2P) is the

dominant traffic in LLN applications. Data is routed towards nodes with some application relevance, such as the LLN gateway to the larger Internet or to the core of private IP networks. In general, these destinations are the DODAG roots and act as data collection points for distributed monitoring applications. Point-to-Multipoint (P2MP) data streams are used for actuation purposes, where messages are sent from DODAG roots to destination nodes. Point-to-Point (P2P) traffic allows communication between two devices belonging to the same LLN, such as a sensor and an actuator. A packet flows from the source to the common ancestor of those two communicating devices, then downward towards the destination. Therefore, RPL has to discover both upward routes (i.e., from nodes to DODAG roots) in order to enable MP2P and P2P flows and downward routes (i.e., from DODAG roots to nodes) to support P2MP and P2P traffic.

Section 3 highlights the challenges that need to be addressed to use RPL on top of TSCH.

Open-source initiatives have emerged around TSCH, with the OpenWSN project [OpenWSN] [OpenWSNETT] being the first open-source implementation of a standards-based protocol stack. This implementation was used as the foundation for an IP for the Smart Objects Alliance (IPSO) [IPSO] interoperability event in 2011. In the absence of a standardized scheduling mechanism for TSCH, a "slotted Aloha" schedule was used.

## 3. Problems and Goals

As highlighted in Appendix A, TSCH differs from other low-power MAC protocols because of its scheduled nature. TSCH defines the mechanisms to execute a communication schedule; yet, it is the entity that sets up the schedule that controls the topology of the network. This scheduling entity also controls the resources allocated to each link in that topology.

How this entity should operate is out of scope of TSCH. The remainder of this section highlights the problems this entity needs to address. For simplicity, we refer to this entity by the generic name "LLC". Note that the 6top sublayer, currently being defined in [SUBLAYER-6top], can be seen as an embodiment of this generic "LLC".

Some of the issues the LLC needs to target might overlap with the scope of other protocols (e.g., 6LoWPAN, RPL, and RSVP). In this case, the LLC will profit from the services provided by other protocols to pursue these objectives.

### 3.1. Network Formation

The LLC needs to control the way the network is formed, including how new nodes join and how already joined nodes advertise the presence of the network. The LLC needs to:

1. Define the Information Elements included in the Enhanced Beacons (EBs) [IEEE.802.15.4e] advertising the presence of the network.

2. (For a new node), define rules to process and filter received EBs.

3. Define the joining procedure. This might include a mechanism to assign a unique 16-bit address to a node and the management of initial keying material.

4. Define a mechanism to secure the joining process and the subsequent optional process of scheduling more communication cells.

### 3.2. Network Maintenance

Once a network is formed, the LLC needs to maintain the network’s health, allowing for nodes to stay synchronized. The LLC needs to:

1. Manage each node’s time source neighbor.

2. Define a mechanism for a node to update the join priority it announces in its EB.

3. Schedule transmissions of EBs to advertise the presence of the network.

### 3.3. Multi-Hop Topology

RPL, given a weighted connectivity graph, determines multi-hop routes. The LLC needs to:

1. Define a mechanism to gather topological information, node and link state, which it can then feed to RPL.

2. Ensure that the TSCH schedule contains cells along the multi-hop routes identified by RPL (a cell in a TSCH schedule is an atomic "unit" of resource, see Section 3.5).

3. Where applicable, maintain independent sets of cells to transport independent flows of data.

### 3.4. Routing and Timing Parents

At all times, a TSCH node needs to have a time-source neighbor to which it can synchronize. Therefore, LLC needs to assign a time-source neighbor to allow for correct operation of the TSCH network. A time-source neighbor could, or not, be taken from the RPL routing parent set.

### 3.5. Resource Management

A cell in a TSCH schedule is an atomic "unit" of resource. The number of cells to assign between neighbor nodes needs to be appropriate for the size of the traffic flow. The LLC needs to:

1. Define a mechanism for neighbor nodes to exchange information about their schedule and, if applicable, negotiate the addition/deletion of cells.

2. Allow for an entity (e.g., a set of devices, a distributed protocol, a Path Computation Element (PCE), etc.) to take control of the schedule.

### 3.6. Dataflow Control

TSCH defines mechanisms for a node to signal when it cannot accept an incoming packet. It does not, however, define the policy that determines when to stop accepting packets. The LLC needs to:

1. Allow for the implementation and configuration of policy to queue incoming and outgoing packets.

2. Manage the buffer space, and indicate to TSCH when to stop accepting incoming packets.

3. Handle transmissions that have failed. A transmission is declared failed when TSCH has retransmitted the packet multiple times, without receiving an acknowledgment. This covers both dedicated and shared cells.

### 3.7. Deterministic Behavior

As highlighted in [RFC5673], in some applications, data is generated periodically and has a well-understood data bandwidth requirement, which is deterministic and predictable. The LLC needs to:

1. Ensure that the data is delivered to its final destination before a deadline possibly determined by the application.

2. Provide a mechanism for such deterministic flows to coexist with bursty or infrequent traffic flows of different priorities.

### 3.8. Scheduling Mechanisms

Several scheduling mechanisms can be envisioned and could possibly coexist in the same network. For example, [RPL] describes how the allocation of bandwidth can be optimized by an external PCE [RFC4655]. Another centralized (PCE-based) traffic-aware scheduling algorithm is defined in [TASA-PIMRC]. Alternatively, two neighbor nodes can adapt the number of cells autonomously by monitoring the amount of traffic and negotiating the allocation to extra cell when needed. An example of a decentralized algorithm (i.e., no PCE is needed) is provided in [Tinka10decentralized]. This mechanism can be used to establish multi-hop paths in a fashion similar to RSVP [RFC2205]. The LLC needs to:

1. Provide a mechanism for two devices to negotiate the allocation and deallocation of cells between them.

2. Provide a mechanism for the device to monitor and manage the capabilities of a node several hops away.

3. Define a mechanism for these different scheduling mechanisms to coexist in the same network.

### 3.9. Secure Communication

Given some keying material, TSCH defines mechanisms to encrypt and authenticate MAC frames. It does not define how this keying material is generated. The LLC needs to:

1. Define the keying material and authentication mechanism needed by a new node to join an existing network.

2. Define a mechanism to allow for the secure transfer of application data between neighbor nodes.

3. Define a mechanism to allow for the secure transfer of signaling data between nodes and the LLC.

## 4. Security Considerations

This memo is an informational overview of existing standards and does not define any new mechanisms or protocols.

It does describe the need for the 6TiSCH WG to define a secure solution. In particular, Section 3.1 describes security in the join process. Section 3.9 discusses data-frame protection.

## 5. References

### 5.1. Normative References

[IEEE.802.15.4]
IEEE, "IEEE Standard for Local and metropolitan area networks -- Part. 15.4: Low-Rate Wireless Personal Area Networks", IEEE Std. 802.15.4-2011, September 2011.

[IEEE.802.15.4e]
IEEE, "IEEE Standard for Local and metropolitan area networks -- Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer", IEEE Std. 802.15.4e-2012, April 2012.

### 5.2. Informative References

[CurrentCalculator]
Linear Technology, "Application Note: Using the Current Calculator to Estimate Mote Power", August 2012, <http://www.linear.com/docs/43189>.

[Doherty07channel]
Doherty, L., Lindsay, W., and J. Simon, "Channel-Specific Wireless Sensor Network Path Data", IEEE International Conference on Computer Communications and Networks (ICCCN), pp. 89-94, 2007.

[IPSO] IPSO Alliance, "IP for Smart Objects Alliance Homepage", <http://www.ipso-alliance.org/>.

[OpenWSN] "Berkeley’s OpenWSN Project Homepage", <http://www.openwsn.org/>.

[OpenWSNETT]
Watteyne, T., Vilajosana, X., Kerkez, B., Chraim, F., Weekly, K., Wang, Q., Glaser, S., and K. Pister, "OpenWSN: A Standards-Based Low-Power Wireless Development Environment", Transactions on Emerging Telecommunications Technologies, Volume 23: Issue 5, August 2012.

[Palattella12standardized]
Palattella, MR., Accettura, N., Vilajosana, X., Watteyne, T., Grieco, LA., Boggia, G., and M. Dohler, "Standardized Protocol Stack For The Internet Of (Important) Things", IEEE Communications Surveys and Tutorials, Volume: 15, Issue 3, December 2012.

[RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <http://www.rfc-editor.org/info/rfc2205>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <http://www.rfc-editor.org/info/rfc2460>.

[RFC3819] Karn, P., Ed., Bormann, C., Fairhurst, G., Grossman, D., Ludwig, R., Mahdavi, J., Montenegro, G., Touch, J., and L. Wood, "Advice for Internet Subnetwork Designers", BCP 89, RFC 3819, DOI 10.17487/RFC3819, July 2004, <http://www.rfc-editor.org/info/rfc3819>.

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <http://www.rfc-editor.org/info/rfc4655>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <http://www.rfc-editor.org/info/rfc4862>.

[RFC4919] Kushalnagar, N., Montenegro, G., and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", RFC 4919, DOI 10.17487/RFC4919, August 2007, <http://www.rfc-editor.org/info/rfc4919>.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <http://www.rfc-editor.org/info/rfc4944>.

[RFC5548] Dohler, M., Ed., Watteyne, T., Ed., Winter, T., Ed., and D. Barthel, Ed., "Routing Requirements for Urban Low-Power and Lossy Networks", RFC 5548, DOI 10.17487/RFC5548, May 2009, <http://www.rfc-editor.org/info/rfc5548>.

[RFC5673] Pister, K., Ed., Thubert, P., Ed., Dwars, S., and T. Phinney, "Industrial Routing Requirements in Low-Power and Lossy Networks", RFC 5673, DOI 10.17487/RFC5673, October 2009, <http://www.rfc-editor.org/info/rfc5673>.

[RFC5826] Brandt, A., Buron, J., and G. Porcu, "Home Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5826, DOI 10.17487/RFC5826, April 2010, <http://www.rfc-editor.org/info/rfc5826>.

[RFC5867] Martocci, J., Ed., De Mil, P., Riou, N., and W. Vermeylen, "Building Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5867, DOI 10.17487/RFC5867, June 2010, <http://www.rfc-editor.org/info/rfc5867>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <http://www.rfc-editor.org/info/rfc6282>.

[RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <http://www.rfc-editor.org/info/rfc6550>.

[RFC6606] Kim, E., Kaspar, D., Gomez, C., and C. Bormann, "Problem Statement and Requirements for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing", RFC 6606, DOI 10.17487/RFC6606, May 2012, <http://www.rfc-editor.org/info/rfc6606>.

[RFC7102] Vasseur, JP., "Terms Used in Routing for Low-Power and Lossy Networks", RFC 7102, DOI 10.17487/RFC7102, January 2014, <http://www.rfc-editor.org/info/rfc7102>.

[RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, DOI 10.17487/RFC7228, May 2014, <http://www.rfc-editor.org/info/rfc7228>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <http://www.rfc-editor.org/info/rfc7252>.

[RPL] Phinney, T., Thubert, P., and R. Assimiti, "RPL applicability in industrial networks", Work in Progress, draft-ietf-roll-rpl-industrial-applicability-02, October 2013.

[SUBLAYER-6top] Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top)", Work in Progress, draft-wang-6tisch-6top-sublayer-01, July 2014.

[TASA-PIMRC] Palattella, MR., Accettura, N., Dohler, M., Grieco, LA., and G. Boggia, "Traffic Aware Scheduling Algorithm for reliable low-power multi-hop IEEE 802.15.4e networks", IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pp. 327-332, September 2012.

[TERMS-6TISCH] Palattella, M., Thubert, P., Watteyne, T., and Q. Wang, "Terminology in IPv6 over the TSCH mode of IEEE 802.15.4e", Work in Progress, draft-ietf-6tisch-terminology-04, March 2015.

[Tinka10decentralized] Tinka, A., Watteyne, T., and K. Pister, "A Decentralized Scheduling Algorithm for Time Synchronized Channel Hopping", Ad Hoc Networks, 2010.

[Watteyne09reliability] Watteyne, T., Mehta, A., and K. Pister, "Reliability Through Frequency Diversity: Why Channel Hopping Makes Sense", Proceedings of the 6th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN), pp. 116-123, October 2009.

## Appendix A. TSCH Protocol Highlights

This appendix gives an overview of the key features of the IEEE 802.15.4e TSCH amendment. It makes no attempt at repeating the standard, rather it focuses on the following:

- Concepts that are sufficiently different from other IEEE 802.15.4 networking that they may need to be defined and presented precisely.
- Techniques and ideas that are part of IEEE 802.15.4e and that might be useful for the work of the 6TiSCH WG.

### A.1. Time Slots

All nodes in a TSCH network are 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 node A to node B, and for node B to reply with an acknowledgment (ACK) frame indicating successful reception.

The duration of a time slot is not defined by the standard. With radios that are compliant with IEEE 802.15.4 operating in the 2.4 GHz frequency band, a maximum-length frame of 127 bytes takes about 4 ms to transmit; a shorter ACK takes about 1 ms. With a 10 ms slot (a typical duration), this leaves 5 ms to radio turnaround, packet processing, and security operations.

### A.2. Slotframes

Time slots are grouped into one of more slotframes. A slotframe continuously repeats over time. TSCH does not impose a slotframe size. Depending on the application needs, these can range from 10's to 1000's of time slots. The shorter the slotframe, the more often a time slot repeats, resulting in more available bandwidth, but also in a higher power consumption.

### A.3. Node TSCH Schedule

A TSCH schedule instructs each node what to do in each time slot: transmit, receive, or sleep. The schedule indicates, for each scheduled (transmit or receive) cell, a channelOffset and the address of the neighbor with which to communicate.

Once a node obtains its schedule, it executes it:

- For each transmit cell, the node checks whether there is a packet in the outgoing buffer that matches the neighbor written in the schedule information for that time slot. If there is none, the node keeps its radio off for the duration of the time slot. If there is one, the node can ask for the neighbor to acknowledge it, in which case it has to listen for the acknowledgment after transmitting.

- For each receive cell, the node listens for possible incoming packets. If none is received after some listening period, it shuts down its radio. If a packet is received, addressed to the node, and passes security checks, the node can send back an acknowledgment.

How the schedule is built, updated, and maintained, and by which entity, is outside of the scope of the IEEE 802.15.4e standard.

### A.4. Cells and Bundles

Assuming the schedule is well built, if node A is scheduled to transmit to node B at slotOffset 5 and channelOffset 11, node B will be scheduled to receive from node A at the same slotOffset and channelOffset.

A single element of the schedule characterized by a slotOffset and channelOffset, and reserved for node A to transmit to node B (or for node B to receive from node A) within a given slotframe, is called a "scheduled cell".

If there is a lot of data flowing from node A to node B, the schedule might contain multiple cells from A to B, at different times. Multiple cells scheduled to the same neighbor can be equivalent, i.e., the MAC layer sends the packet on whichever of these cells shows up first after the packet was put in the MAC queue. The union of all cells between two neighbors, A and B, is called a "bundle". Since the slotframe repeats over time (and the length of the slotframe is typically constant), each cell gives a "quantum" of bandwidth to a given neighbor. Modifying the number of equivalent cells in a bundle modifies the amount of resources allocated between two neighbors.

### A.5. Dedicated vs. Shared Cells

By default, each scheduled transmit cell within the TSCH schedule is dedicated, i.e., reserved only for node A to transmit to node B. IEEE 802.15.4e also allows a cell to be marked as shared. In a shared cell, multiple nodes can transmit at the same time, on the same frequency. To avoid contention, TSCH defines a backoff algorithm for shared cells.

A scheduled cell can be marked as both transmitting and receiving. In this case, a node transmits if it has an appropriate packet in its output buffer, or listens otherwise. Marking a cell as [transmit,receive,shared] results in slotted-Aloha behavior.

### A.6. Absolute Slot Number

TSCH defines a timeslot counter called Absolute Slot Number (ASN). When a new network is created, the ASN is initialized to 0; from then on, it increments by 1 at each timeslot. In detail:

ASN = (k*S+t)

where k is the slotframe cycle (i.e., the number of slotframe repetitions since the network was started), S the slotframe size, and t the slotOffset. A node learns the current ASN when it joins the network. Since nodes are synchronized, they all know the current value of the ASN, at any time. The ASN is encoded as a 5-byte number: this allows it to increment for hundreds of years (the exact value depends on the duration of a timeslot) without wrapping over. The ASN is used to calculate the frequency to communicate on and can be used for security-related operations.

### A.7. Channel Hopping

For each scheduled cell, the schedule specifies a slotOffset and a channelOffset. In a well-built schedule, when node A has a transmit cell to node B on channelOffset 5, node B has a receive cell from node A on the same channelOffset. The channelOffset is translated by both nodes into a frequency using the following function:

frequency = F {(ASN + channelOffset) mod nFreq}

The function F consists of a lookup table containing the set of available channels. The value nFreq (the number of available frequencies) is the size of this lookup table. There are as many channelOffset values as there are frequencies available (e.g., 16 when using radios that are compliant with IEEE 802.15.4 at 2.4 GHz, when all channels are used). Since both nodes have the same

channelOffset written in their schedule for that scheduled cell, and the same ASN counter, they compute the same frequency. At the next iteration (cycle) of the slotframe, however, while the channelOffset is the same, the ASN has changed, resulting in the computation of a different frequency.

This results in "channel hopping": even with a static schedule, pairs of neighbors "hop" between the different frequencies when communicating. A way of ensuring communication happens on all available frequencies is to set the number of timeslots in a slotframe to a prime number. Channel hopping is a technique known to efficiently combat multi-path fading and external interference [Watteyne09reliability].

### A.8. Time Synchronization

Because of the slotted nature of communication in a TSCH network, nodes have to maintain tight synchronization. All nodes are assumed to be equipped with clocks to keep track of time. Yet, because clocks in different nodes drift with respect to one another, neighbor nodes need to periodically resynchronize.

Each node needs to periodically synchronize its network clock to another node, and it also provides its network time to its neighbors. It is up to the entity that manages the schedule to assign an adequate time source neighbor to each node, i.e., to indicate in the schedule which neighbor is its "time source neighbor". While setting the time source neighbor, it is important to avoid synchronization loops, which could result in the formation of independent clusters of synchronized nodes.

TSCH adds timing information in all packets that are exchanged (both data and ACK frames). This means that neighbor nodes can resynchronize to one another whenever they exchange data. In detail, two methods are defined in IEEE 802.15.4e (of 2012) for allowing a device to synchronize in a TSCH network: (i) Acknowledgment-based and (ii) Frame-based synchronization. In both cases, the receiver calculates the difference in time between the expected time of frame arrival and its actual arrival. In Acknowledgment-based synchronization, the receiver provides such information to the sender node in its acknowledgment. In this case, it is the sender node that synchronizes to the clock of the receiver. In Frame-based synchronization, the receiver uses the computed delta for adjusting its own clock. In this case, it is the receiver node that synchronizes to the clock of the sender.

Different synchronization policies are possible. Nodes can keep synchronization exclusively by exchanging EBs. Nodes can also keep synchronized by periodically sending valid frames to a time source neighbor and use the acknowledgment to resynchronize. Both methods (or a combination thereof) are valid synchronization policies; which one to use depends on network requirements.

### A.9. Power Consumption

There are only a handful of activities a node can perform during a timeslot: transmit, receive, or sleep. Each of these operations has some energy cost associated to them; the exact value depends on the hardware used. Given the schedule of a node, it is straightforward to calculate the expected average power consumption of that node.

### A.10. Network TSCH Schedule

The schedule entirely defines the synchronization and communication between nodes. By adding/removing cells between neighbors, one can adapt a schedule to the needs of the application. Intuitive examples are:

- Make the schedule "sparse" for applications where nodes need to consume as little energy as possible, at the price of reduced bandwidth.
- Make the schedule "dense" for applications where nodes generate a lot of data, at the price of increased power consumption.
- Add more cells along a multi-hop route over which many packets flow.

### A.11. Join Process

Nodes already part of the network can periodically send EB frames to announce the presence of the network. These contain information about the size of the timeslot used in the network, the current ASN, information about the slotframes and timeslots the beaconing node is listening on, and a 1-byte join priority. The join priority field gives information to make a better decision of which node to join. Even if a node is configured to send all EB frames on the same channelOffset, because of the channel hopping nature of TSCH described in Appendix A.7, this channelOffset translates into a different frequency at different slotframe cycles. As a result, EB frames are sent on all frequencies.

A node wishing to join the network listens for EBs. Since EBs are sent on all frequencies, the joining node can listen on any frequency until it hears an EB. What frequency it listens on is implementation specific. Once it has received one or more EBs, the new node enables the TSCH mode and uses the ASN and the other timing information from the EB to synchronize to the network. Using the slotframe and cell information from the EB, it knows how to contact other nodes in the network.

The IEEE 802.15.4e TSCH standard does not define the steps beyond this network "bootstrap".

### A.12. Information Elements

TSCH introduces the concept of Information Elements (IEs). An IE is a list of Type-Length-Value containers placed at the end of the MAC header. A small number of types are defined for TSCH (e.g., the ASN in the EB is contained in an IE), and an unmanaged range is available for extensions.

A data bit in the MAC header indicates whether the frame contains IEs. IEs are grouped into Header IEs, consumed by the MAC layer and therefore typically invisible to the next higher layer, and Payload IEs, which are passed untouched to the next higher layer, possibly followed by regular payload. Payload IEs can therefore be used for the next higher layers of two neighbor nodes to exchange information.

### A.13. Extensibility

The TSCH standard is designed to be extensible. It introduces the mechanisms as "building block" (e.g., cells, bundles, slotframes, etc.), but leaves entire freedom to the upper layer to assemble those. The MAC protocol can be extended by defining new Header IEs. An intermediate layer can be defined to manage the MAC layer by defining new Payload IEs.

## Appendix B. TSCH Features

This section details features of TSCH, which might be interesting for the work of the 6TiSCH WG. It does not define any requirements.

### B.1. Collision-Free Communication

TSCH allows one to design a schedule that yields collision-free communication. This is done by building the schedule with dedicated cells in such a way that at most, one node communicates with a specific neighbor in each slotOffset/channelOffset cell. Multiple pairs of neighbor nodes can exchange data at the same time, but on different frequencies.

### B.2. Multi-Channel vs. Channel Hopping

A TSCH schedule looks like a matrix of width "slotframe size", S, and of height "number of frequencies", nFreq. For a scheduling algorithm, cells can be considered atomic "units" to schedule. In particular, because of the channel hopping nature of TSCH, the scheduling algorithm should not worry about the actual frequency communication happens on, since it changes at each slotframe iteration.

### B.3. Cost of (Continuous) Synchronization

When there is traffic in the network, nodes that are communicating implicitly resynchronize using the data frames they exchange. In the absence of data traffic, nodes are required to synchronize to their time source neighbor(s) periodically not to drift in time. If they have not been communicating for some time (typically 30 s), nodes can exchange a dummy data frame to resynchronize. The frequency at which such messages need to be transmitted depends on the stability of the clock source and on how "early" each node starts listening for data (the "guard time"). Theoretically, with a 10 ppm clock and a 1 ms guard time, this period can be 100 s. Assuming this exchange causes the node’s radio to be on for 5 ms, this yields a radio duty cycle needed to keep synchronized of 5 ms / 100 s = 0.005%. While TSCH does require nodes to resynchronize periodically, the cost of doing so is very low.

### B.4. Topology Stability

The channel hopping nature of TSCH causes links to be very "stable". Wireless phenomena such as multi-path fading and external interference impact a wireless link between two nodes differently on each frequency. If a transmission from node A to node B fails, retransmitting on a different frequency has a higher likelihood of

succeeding that retransmitting on the same frequency. As a result, even when some frequencies are "behaving bad", channel hopping "smoothens" the contribution of each frequency, resulting in more stable links and therefore a more stable topology.

### B.5. Multiple Concurrent Slotframes

The TSCH standard allows for multiple slotframes to coexist in a node's schedule. It is possible that, at some timeslot, a node has multiple activities scheduled (e.g., transmit to node B on slotframe 2, receive from node C on slotframe 1). To handle this situation, the TSCH standard defines the following precedence rules:

1. Transmissions take precedence over receptions;

2. Lower slotframe identifiers take precedence over higher slotframe identifiers.

In the example above, the node would transmit to node B on slotframe 2.

## Acknowledgments

Special thanks to Dominique Barthel, Patricia Brett, Guillaume Gaillard, Pat Kinney, Ines Robles, Timothy J. Salo, Jonathan Simon, Rene Struik, and Xavi Vilajosana for reviewing the document and providing valuable feedback. Thanks to the IoT6 European Project (STREP) of the 7th Framework Program (Grant 288445).

Authors' Addresses

Thomas Watteyne (editor)
Linear Technology
32990 Alvarado-Niles Road, Suite 910
Union City, CA 94587
United States

Phone: +1 (510) 400-2978
EMail: twatteyne@linear.com

Maria Rita Palattella
University of Luxembourg
Interdisciplinary Centre for Security, Reliability and Trust
4, rue Alphonse Weicker
Luxembourg L-2721
Luxembourg

Phone: +352 46 66 44 5841
EMail: maria-rita.palattella@uni.lu

Luigi Alfredo Grieco
Politecnico di Bari
Department of Electrical and Information Engineering
Via Orabona 4
Bari 70125
Italy

Phone: +39 08 05 96 3911
EMail: a.grieco@poliba.it