

6TSCH  
Internet-Draft  
Intended status: Informational  
Expires: August 12, 2013

T. Watteyne, Ed.  
Linear Technology  
February 8, 2013

**Using IEEE802.15.4e TSCH in an LLN context:  
Overview, Problem Statement and Goals  
draft-watteyne-6tsch-tsch-lln-context-00**

Abstract

This document describes the environment, problem statement, and goals for using the IEEE802.15.4e TSCH MAC protocol in the context of LLNs. The set of goals enumerated in this document form an initial set only.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 12, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction . . . . .</a>	<a href="#">3</a>
<a href="#">2.</a>	<a href="#">TSCH in the LLN Context . . . . .</a>	<a href="#">5</a>
<a href="#">3.</a>	<a href="#">Problems and Goals . . . . .</a>	<a href="#">7</a>
<a href="#">3.1.</a>	<a href="#">Network Formation . . . . .</a>	<a href="#">7</a>
<a href="#">3.2.</a>	<a href="#">Network Maintenance . . . . .</a>	<a href="#">7</a>
<a href="#">3.3.</a>	<a href="#">Multi-Hop Topology . . . . .</a>	<a href="#">8</a>
<a href="#">3.4.</a>	<a href="#">Routing and Timing Parents . . . . .</a>	<a href="#">8</a>
<a href="#">3.5.</a>	<a href="#">Resource Management . . . . .</a>	<a href="#">8</a>
<a href="#">3.6.</a>	<a href="#">Dataflow Control . . . . .</a>	<a href="#">9</a>
<a href="#">3.7.</a>	<a href="#">Secure Communication . . . . .</a>	<a href="#">9</a>
<a href="#">4.</a>	<a href="#">Contributors . . . . .</a>	<a href="#">10</a>
<a href="#">5.</a>	<a href="#">Acknowledgements . . . . .</a>	<a href="#">11</a>
<a href="#">6.</a>	<a href="#">References . . . . .</a>	<a href="#">12</a>
<a href="#">6.1.</a>	<a href="#">Normative References . . . . .</a>	<a href="#">12</a>
<a href="#">6.2.</a>	<a href="#">Informative References . . . . .</a>	<a href="#">12</a>
<a href="#">6.3.</a>	<a href="#">External Informative References . . . . .</a>	<a href="#">14</a>
<a href="#">Appendix A.</a>	<a href="#">TSCH Protocol Highlights . . . . .</a>	<a href="#">18</a>
<a href="#">A.1.</a>	<a href="#">Timeslots . . . . .</a>	<a href="#">18</a>
<a href="#">A.2.</a>	<a href="#">Slotframes . . . . .</a>	<a href="#">18</a>
<a href="#">A.3.</a>	<a href="#">Mote Communication Schedule . . . . .</a>	<a href="#">18</a>
<a href="#">A.4.</a>	<a href="#">Links and Paths . . . . .</a>	<a href="#">19</a>
<a href="#">A.5.</a>	<a href="#">Dedicated vs. Shared Slots . . . . .</a>	<a href="#">19</a>
<a href="#">A.6.</a>	<a href="#">Absolute Slot Number . . . . .</a>	<a href="#">20</a>
<a href="#">A.7.</a>	<a href="#">Channel Hopping . . . . .</a>	<a href="#">20</a>
<a href="#">A.8.</a>	<a href="#">Time Synchronization . . . . .</a>	<a href="#">21</a>
<a href="#">A.9.</a>	<a href="#">Power Consumption . . . . .</a>	<a href="#">21</a>
<a href="#">A.10.</a>	<a href="#">Network Communication Schedule . . . . .</a>	<a href="#">21</a>
<a href="#">A.11.</a>	<a href="#">Join Process . . . . .</a>	<a href="#">22</a>
<a href="#">A.12.</a>	<a href="#">Information Elements . . . . .</a>	<a href="#">22</a>
<a href="#">A.13.</a>	<a href="#">Extensibility . . . . .</a>	<a href="#">23</a>
<a href="#">Appendix B.</a>	<a href="#">TSCH Gotchas . . . . .</a>	<a href="#">24</a>
<a href="#">B.1.</a>	<a href="#">Collision Free Communication . . . . .</a>	<a href="#">24</a>
<a href="#">B.2.</a>	<a href="#">Multi-Channel vs. Channel Hopping . . . . .</a>	<a href="#">24</a>
<a href="#">B.3.</a>	<a href="#">Cost of (continuous) Synchronization . . . . .</a>	<a href="#">24</a>
<a href="#">B.4.</a>	<a href="#">Topology Stability . . . . .</a>	<a href="#">24</a>
<a href="#">B.5.</a>	<a href="#">Multiple Concurrent Slotframes . . . . .</a>	<a href="#">25</a>
	<a href="#">Author's Address . . . . .</a>	<a href="#">26</a>



## **1. Introduction**

The IEEE802.15.4e standard [[IEEE802154e](#)] was published in 2012 as an amendment to the Medium Access Control (MAC) protocol of the IEEE802.15.4-2011 [[IEEE802154](#)] standard. The Timeslotted Channel Hopping (TSCH) mode of the IEEE802.15.4e standard is the object of this document.

TSCH was designed to "allow IEEE802.15.4 devices to support a wide range of industrial applications" [[IEEE802154e](#)]. 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. This is very different from the "legacy" IEEE802.15.4 MAC protocol, and is therefore better described as a "redesign". TSCH does not amend the physical layer; i.e. it can operate on any IEEE802.15.4-compliant hardware.

IEEE802.15.4e can be seen as the latest generation of ultra-lower power and reliable networking solutions for LLNs. Its core technology is similar to the one used in industrial networking technologies such as WirelessHART [[WHART](#)] or ISA100.11a [[ISA100](#)]. These protocol solutions have been targeted essentially at the industrial market. WirelessHART is for example the wireless extension of HART, a long standing protocol suite for networking industrial equipment.

[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. Industrial protocols such as WirelessHART satisfy those requirements, and with tens of thousands of networks deployed [[Emerson](#)], these types of networks have a large impact on industrial applications. Commercial networking solutions are available today in which motes consume 10's of micro-amps on average [[CurrentCalculator](#)] with end-to-end packet delivery ratios over 99.999% [Doherty].

IEEE802.15.4e builds on the same foundations as WirelessHART, and therefore exhibits similar performance. Yet, unlike an industrial protocol which is, by nature, application-specific, 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, running 6LoWPAN [[RFC6282](#)], RPL [[RFC6550](#)] and CoAP [[I-D.ietf-core-coap](#)].

Bringing industrial-like performance into the LLN stack developed by the 6LoWPAN, ROLL and CORE working groups will open 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](#)].

While [[IEEE802154e](#)] defines the mechanisms for a TSCH mote 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-configuring IP addresses, or manage keying material.



## **2. TSCH in the LLN Context**

In many cases, to map the services required by the IP layer to the services provided by the link layer, an adaptation layer is used [[palattella12standardized](#)]. The 6LoWPAN working group has started in 2007 to work on specifications for transmitting IPv6 over IEEE802.15.4 networks [[RFC4919](#)]. Typically, low-power WPANs are characterized by small packet sizes, support for addresses with different lengths, low bandwidth, star and mesh topologies, battery powered devices, low cost, large number of devices, unknown node positions, high unreliability, and periods during which communication interfaces are turned off to save 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 un-fragmented IPv6 packet would be too large to fit in an IEEE802.15.4 frame. Moreover, the overhead due to the 40-byte long IPv6 header would waste the scarce bandwidth available at the PHY layer [[RFC4944](#)]. For these reasons, the 6LoWPAN working group has defined an effective adaptation layer [[RFC6568](#)]. Further issues encompass the auto-configuration of IPv6 addresses [[RFC2464](#)][[RFC6755](#)], 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 supplied 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 working group 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](#)]. It is able to quickly build up network routes, to distribute routing knowledge among nodes, and to adapt the topology. In a typical setting, motes are connected through multi-hop paths to a small set of root devices, which are usually responsible for data collection and coordination duties. 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 destination, following a gradient-based approach. RPL encompasses 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 they 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, e.g. a sensor and an actuator. A packet will flow from the source towards the common ancestor of those two communicating devices, then downward towards the destination. As a consequence, 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.

Several open-source initiatives have emerged around TSCH. The OpenWSN project [[OpenWSN](#)][OpenWSNETT] is an open-source implementation of a fully standards-based protocol stack, which aims at evaluating the applicability of TSCH to different applications. This implementation was used as the foundation for an IP for Smart Objects Alliance (IPSO) [[IPSO](#)] interoperability demo 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 is different for traditional low-power MAC protocols because of its scheduled nature. TSCH defines the mechanisms to execute a communication schedule, but it is the entity that sets up that schedule which controls the topology of the network, and 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 will refer to this entity by the generic name "6TSCH", without loss of generality. In particular, we do not assume the nature of 6TSCH, whether an adaptation layer, a distributed reservation protocol, a centralized path computation engine, or any combination thereof.

#### **3.1. Network Formation**

6TSCH needs to control the way the network is formed, including how new motes join, and how already joined motes advertise the presence of the network. 6TSCH needs to:

1. Define the Information Elements to include in Enhanced Beacons advertising the presence of the network.
2. For a new mote, define rules to process and filter received Enhanced Beacons. This includes a mechanism to select the best mote to join the network through.
3. Define the joining procedure. This includes a mechanism to assign a unique 16-bit address to a mote, and the management of initial keying material.

#### **3.2. Network Maintenance**

Once a network is formed, 6TSCH needs to maintain the network's health, allowing for motes to stay synchronized. 6TSCH needs to:

1. Manage each mote's time source neighbor(s).
2. Define a mechanism for a mote to update the join priority it announces in its Enhanced Beacon.
3. Schedule transmissions of an Enhanced Beacon to advertise the presence of the network.



### **3.3. Multi-Hop Topology**

RPL, given a weighted connectivity graph, determines multi-hop routes. 6TSCH needs to:

1. Define a mechanism whereby it gathers topological information, which it can then feed to RPL.
2. Ensure that the TSCH schedule contains links along the multi-hop routes identified by RPL.
3. Where applicable, maintain independent sets of links to transport independent flows of data.

### **3.4. Routing and Timing Parents**

At all times, a TSCH mote needs to have at least one time source neighbor it can synchronize to. 6TSCH therefore needs to assign time source neighbors to allow for correct operation of the TSCH network. These time source neighbors could, or not, be related to RPL time parents.

### **3.5. Resource Management**

A link in a TSCH schedule is a "unit" of resource. The number of links to assign between neighbor motes needs to be appropriate for the size of the traffic flow. 6TSCH needs to:

1. Define rules on when to create or delete a slotframes.
2. Define rules to determine the length of a slotframe, and the trigger to modify the length of a slotframe.
3. Define rules on when to add or delete links in a particular slotframe.
4. Define a mechanism for neighbor nodes to exchange information about their schedule and, if applicable, negotiate the addition/deletion of links.
5. Allow for a (possibly centralized) entity to take full control over the schedule.
6. Define a set of metrics to evaluate the tradeoff between latency, bandwidth and energy consumption achieved by a particular schedule.



### **3.6. Dataflow Control**

TSCH defines mechanisms for a mote to signal it cannot accept an incoming packet. It does not, however, define the policy which determines when not to accept. 6TSCH need to:

1. Define a queueing policy for incoming and outgoing packets.
2. Manage the buffer space and indicate to TSCH when to stop accepting incoming packet.
3. Handle transmissions that have failed.

### **3.7. 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. 6TSCH need to:

1. Define the keying material and authentication mechanism needed by a new mote to join an existing network.
2. Define a mechanism to allow for the secure transfer of application data between neighbor motes.
3. Define a mechanism to allow for the secure transfer of signaling data between motes and 6TSCH.





#### **4. Contributors**

Maria Rita Palattella  
SnT/University of Luxembourg

maria-rita.palattella@uni.lu

Luigi Alfredo Grieco  
Politecnico di Bari

a.grieco@poliba.it

## **5. Acknowledgements**

Special thanks to Jonathan Simon for his review and valuable comments.

## **6. References**

### **6.1. Normative References**

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

### **6.2. Informative References**

- [RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", [RFC 2464](#), December 1998.
- [RFC3819] Karn, P., 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](#), July 2004.
- [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](#), August 2007.
- [RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", [RFC 4944](#), September 2007.
- [RFC5548] Dohler, M., Watteyne, T., Winter, T., and D. Barthel, "Routing Requirements for Urban Low-Power and Lossy Networks", [RFC 5548](#), May 2009.
- [RFC5826] Brandt, A., Buron, J., and G. Porcu, "Home Automation Routing Requirements in Low-Power and Lossy Networks", [RFC 5826](#), April 2010.
- [RFC5867] Martocci, J., De Mil, P., Riou, N., and W. Vermeylen, "Building Automation Routing Requirements in Low-Power and Lossy Networks", [RFC 5867](#), June 2010.
- [RFC5673] Pister, K., Thubert, P., Dwars, S., and T. Phinney, "Industrial Routing Requirements in Low-Power and Lossy Networks", [RFC 5673](#), October 2009.
- [RFC6282] Hui, J. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", [RFC 6282](#), September 2011.
- [RFC6550] Winter, T., Thubert, P., 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](#), March 2012.

[RFC6568] Kim, E., Kaspar, D., and JP. Vasseur, "Design and Application Spaces for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", [RFC 6568](#), April 2012.

[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](#), May 2012.

[RFC6755] Campbell, B. and H. Tschofenig, "An IETF URN Sub-Namespaces for OAuth", [RFC 6755](#), October 2012.

[I-D.thubert-roll-forwarding-frags]  
Thubert, P. and J. Hui, "LLN Fragment Forwarding and Recovery", [draft-thubert-roll-forwarding-frags-00](#) (work in progress), March 2012.

[I-D.tsao-roll-security-framework]  
Tsao, T., Alexander, R., Daza, V., and A. Lozano, "A Security Framework for Routing over Low Power and Lossy Networks", [draft-tsao-roll-security-framework-02](#) (work in progress), March 2010.

[I-D.thubert-roll-asymlink]  
Thubert, P., "RPL adaptation for asymmetrical links", [draft-thubert-roll-asymlink-02](#) (work in progress), December 2011.

[I-D.ietf-roll-terminology]  
Vasseur, J., "Terminology in Low power And Lossy Networks", [draft-ietf-roll-terminology-10](#) (work in progress), January 2013.

[I-D.ietf-roll-p2p-rpl]  
Goyal, M., Baccelli, E., Philipp, M., Brandt, A., and J. Martocci, "Reactive Discovery of Point-to-Point Routes in Low Power and Lossy Networks", [draft-ietf-roll-p2p-rpl-16](#) (work in progress), February 2013.

[I-D.ietf-roll-trickle-mcast]  
Hui, J. and R. Kelsey, "Multicast Protocol for Low power and Lossy Networks (MPL)", [draft-ietf-roll-trickle-mcast-03](#) (work in progress), January 2013.



[I-D.thubert-6lowpan-backbone-router]

Thubert, P., "6LoWPAN Backbone Router",  
[draft-thubert-6lowpan-backbone-router-02](#) (work in progress), June 2010.

[I-D.sarikaya-core-sbootstrapping]

Sarikaya, B., Ohba, Y., Moskowitz, R., Cao, Z., and R. Cragie, "Security Bootstrapping Solution for Resource-Constrained Devices",  
[draft-sarikaya-core-sbootstrapping-04](#) (work in progress), April 2012.

[I-D.gilger-smart-object-security-workshop]

Gilger, J. and H. Tschofenig, "Report from the 'Smart Object Security Workshop', 23rd March 2012, Paris, France", [draft-gilger-smart-object-security-workshop-00](#) (work in progress), October 2012.

[I-D.phinney-roll-rpl-industrial-applicability]

Phinney, T., Thubert, P., and R. Assimiti, "RPL applicability in industrial networks",  
[draft-phinney-roll-rpl-industrial-applicability-01](#) (work in progress), October 2012.

[I-D.ietf-core-coap]

Shelby, Z., Hartke, K., Bormann, C., and B. Frank, "Constrained Application Protocol (CoAP)",  
[draft-ietf-core-coap-13](#) (work in progress), December 2012.

### **6.3. External Informative References**

[IEEE802154e]

IEEE standard for Information Technology, "IEEE std. 802.15.4e, Part. 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer", April 2012.

[IEEE802154]

IEEE standard for Information Technology, "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", June 2011.

[WHART]

[www.hartcomm.org](http://www.hartcomm.org), "Highway Addressable Remote Transducer, a group of specifications for industrial process and control devices administered by the HART Foundation".

[ISA100]

ISA, "ISA100, Wireless Systems for Automation", May 2008,





< <http://www.isa.org/Community/SP100WirelessSystemsforAutomation>>.

[Emerson] Emerson Process Management, "Emerson Process Management Smart Wireless Homepage", < <http://www2.emersonprocess.com/en-US/plantweb/wireless/Pages/WirelessHomePage-Flash.aspx>>.

[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 2012, August 2012, <<http://onlinelibrary.wiley.com/doi/10.1002/ett.2558/abstract>>.

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

[CurrentCalculator] Linear Technology, "Application Note: Using the Current Calculator to Estimate Mote Power", August 2012, <[http://cds.linear.com/docs/en/application-note/Application\\_Note\\_-\\_Using\\_the\\_Current\\_Calculator\\_to\\_Estimate\\_Mote\\_Power.pdf](http://cds.linear.com/docs/en/application-note/Application_Note_-_Using_the_Current_Calculator_to_Estimate_Mote_Power.pdf)>.

[doherty07channel] Doherty, L., Lindsay, W., and J. Simon, "Channel-Specific Wireless Sensor Network Path Data", IEEE International Conference on Computer Communications and Networks (ICCCN) 2008, 2007.

[TSMP] Pister, K. and L. Doherty, "TSMP: Time Synchronized Mesh Protocol", International Symposium on Distributed Sensor Networks (DSN) 2008, Nov. 2008, < <http://robotics.eecs.berkeley.edu/~pister/publications/2008/TSMP%20DSN08.pdf>>.

[tinka10decentralized] Tinka, A., Watteyne, T., and K. Pister, "A Decentralized Scheduling Algorithm for Time Synchronized Channel Hopping", Ad Hoc Networks 2010, 2010, < <http://robotics.eecs.berkeley.edu/~pister/publications/2008/TSMP%20DSN08.pdf>>.



## [watteyne09reliability]

Watteyne, T., Mehta, A., and K. Pister, "Reliability Through Frequency Diversity: Why Channel Hopping Makes Sense", International Conference on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN) 2009, Oct. 2009, <[http://www.ietf.org/mail-archive/web/roll/current/pdfa\\_EzmuDiv3.pdf](http://www.ietf.org/mail-archive/web/roll/current/pdfa_EzmuDiv3.pdf)>.

## [kerkez09feasibility]

Kerkez, B., Watteyne, T., and M. Magliocco, "Feasibility analysis of controller design for adaptive channel hopping", International Workshop on Performance Methodologies and Tools for Wireless Sensor Networks (WSNPERF) 2009, Oct. 2009, <[http://www-bsac.eecs.berkeley.edu/publications/search/send\\_publication\\_pdf2client.php?pubID=1249681245](http://www-bsac.eecs.berkeley.edu/publications/search/send_publication_pdf2client.php?pubID=1249681245)>.

## [TASA-PIMRC]

Palattella, MR., Accettura, N., Dohler, M., Grieco, LA., and G. Boggia, "Traffic Aware Scheduling Algorithm for Multi-Hop IEEE 802.15.4e Networks", IEEE PIMRC 2012, Sept. 2012, < <http://www.cttc.es/resources/doc/120531-submitted-tasa-25511.pdf>>.

## [TASA-SENSORS]

Palattella, MR., Accettura, N., Dohler, M., Grieco, LA., and G. Boggia, "Traffic-Aware Time-Critical Scheduling In Heavily Duty-Cycled IEEE 802.15.4e For An Industrial IoT", IEEE SENSORS 2012, Oct. 2012, < <http://www.cttc.es/resources/doc/120821-sensors2012-4396981770946977737.pdf>>.

## [TASA-WCNC]

Accettura, N., Palattella, MR., Dohler, M., Grieco, LA., and G. Boggia, "Standardized Power-Efficient and Internet-Enabled Communication Stack for Capillary M2M Networks", IEEE WCNC 2012, Apr. 2012, < <http://www.cttc.es/resources/doc/120109-1569521283-submitted-58230.pdf>>.

## [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 2012, Dec. 2012, < <http://www.cttc.es/resources/doc/121025-completestackforiot-clean-4818610916636121981.pdf>>.

## [PANA]

Kanda, M., Ohba, Y., Das, S., and S. Chasko, "PANA applicability in constrained environments", Febr. 2012, <h



[ttp://www.lix.polytechnique.fr/hipercom/SmartObjectSecurity/papers/MitsuruKanda.pdf](http://www.lix.polytechnique.fr/hipercom/SmartObjectSecurity/papers/MitsuruKanda.pdf)>.

## **Appendix A. TSCH Protocol Highlights**

This appendix gives an overview of the key features of the IEEE802.15.4e Timeslotted Channel Hopping (TSCH) amendment. It makes no attempt at repeating the standard, but rather focuses on the following:

- o Concepts which are sufficiently different from traditional IEEE802.15.4 networking that they may need to be defined and presented precisely.
- o Techniques and ideas which are part of IEEE802.15.4e and which might be useful for the work of 6TSCH.

### **A.1. Timeslots**

All motes 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 mote A to mote B, and for mote B to reply with an acknowledgment (ACK) frame indicating successful reception.

The duration of a timeslot is not defined by the standard. With IEEE802.15.4-compliant radios operating in the 2.4GHz frequency band, a maximum-length frame of 127 bytes takes about 4ms to transmit; a shorter ACK takes about 1ms. With a 10ms slot (a typical duration), this leaves 5ms to radio turnaround, packet processing and security operations.

### **A.2. Slotframes**

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

### **A.3. Mote Communication Schedule**

A communication schedule instructs each mote what to do in each slot: transmit, receive or sleep. The schedule indicates, for each active (transmit or receive) timeslot a channelOffset and the address of the neighbor to communicate with.

Once a mote obtains its schedule, it executes it:

- o For each transmit slot, the mote checks whether there is a packet in the outgoing buffer which matches the neighbor written in the





schedule information for that slot. If there is none, the mote keeps its radio off for the duration of the slot. If there is one, the mote can ask for the neighbor to acknowledge it, in which case it has to listen for the acknowledgment after the transmission.

- o For each receive slot, the mote 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 mote, and passes security checks, the mote typically sends back and acknowledgment.

How the schedule is built, updated and maintained, and by which entity, is outside of the scope of the IEEE802.15.4e standard.

#### **A.4. Links and Paths**

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

A single slot of the schedule (i.e., a single cell of the grid), characterized by a slotOffset and channelOffset, and reserved for mote A to transmit to mote B (or for mote B to receive from mote A), is called a "link".

If there is a lot of data flowing from mote A to mote B, the schedule might contain multiple slots from A to B, at different times. Multiple links scheduled to the same neighbor are typically equivalent, i.e. the MAC layer sends the packet on whichever of these links happens to show up after the packet was put in the MAC queue. The union of all links between two neighbors, A and B, is called a "path". Since the slotframe repeats over time (and the length of the slotframe is typically constant), each link gives a "quantum" of bandwidth to a given neighbor. Modifying the number of links in a path modify the amount of resources allocated between two neighbors.

#### **A.5. Dedicated vs. Shared Slots**

By default, each transmit timeslot within the TSCH schedule is dedicated, i.e., reserved only for mote A to transmit to mote B. IEEE802.15.4e allows also to mark a slot as shared. In a shared slot, multiple motes can transmit at the same time, on the same frequency. To avoid contention, TSCH defines a back-off algorithm for shared slots.

A slot can be marked as both transmitting and receiving. In this



case, a mote transmits if it has an appropriate packet in its output buffer, or listens otherwise. Marking a slot as [transmit,shared,receive] 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:

$$\text{ASN} = (k \cdot S + t)$$

where S is the slotframe size and k the slotframe cycle (i.e., the number of slotframe occurrences over time). A mote learns the current ASN when it joins the network. Since motes are synchronized, they all know the current value of the ASN, and 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. The ASN is used (i) to calculate the frequency to communicate on, jointly with the channelOffset, (ii) to build unique security nonce counters, when the CCM\* mode is activated.

#### [A.7.](#) Channel Hopping

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

$$\text{frequency} = F \{(\text{ASN} + \text{channelOffset}) \bmod nCh\}$$

The function F consists of a look-up table containing the set of available channels. The value nCh (the number of available frequencies) is the size of this look-up table. 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). Since both motes have the same channelOffset written in their schedule for that timeslot, and the same ASN counter since they are synchronized, they compute the same frequency. At the next iteration (cycle) of the slotframe, however, the channelOffset will be the same, but the ASN will have changed, resulting in the computation of a different frequency. If the slotframe size, S (used for computing ASN), and the number of channel offsets, nCh, are relatively prime, the translation function ensures that each link rotates through k available channels over k slotframe cycles. This results in "channel hopping": even with a static schedule, pairs of neighbors "hop" between the different frequencies when communicating.



The look-up table *F* can be built to contain only a subset of all available channels. This results in "blacklisting".

Channel hopping is a technique known to efficiently combat multi-path fading and external interference. This results in a TSCH network having a more stable topology than if only a single channel were used for the entire network.

#### **A.8. Time Synchronization**

Because of the slotted nature of communication in a TSCH network, motes have to maintain tight synchronization. All motes are assumed to be equipped with clocks to keep track of time (32kHz crystal oscillators are typically used). Yet, because clocks in different motes drift with respect to one another, neighbor motes need to periodically re-synchronize.

TSCH adds timing information in all packets that are exchanged (both data and ACK frames). This means that neighbor motes can resynchronize to one another whenever they exchange data. The schedule of a mote indicates which of its neighbor(s) are its "time source neighbors". When a mote communicates with one of its time source neighbors, it uses the timing information in the exchanged frames to realign its clock to its neighbor's.

It is up to the entity that manages the schedule to assign adequate time source neighbor(s) to each mote. It is important to avoid synchronization loops, which could result in the formation of independent clusters of motes.

#### **A.9. Power Consumption**

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

#### **A.10. Network Communication Schedule**

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

- o Make the schedule "sparse" for applications where motes need to consume as little as possible, at the price of reduced bandwidth.



- o Make the schedule "dense" for applications where motes generate a lot of data, at the price of increased power consumption.
- o Add more links along a multi-hop route over which many packets flow.

#### **A.11. Join Process**

Motes already part of the network can periodically send Enhanced Beacon (EB) frames to announce the presence 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 mote is listening on, and a 1-byte join priority. This join priority corresponds to the number of hops separating the mote sending the EB, and the PAN coordinator. Because of the channel hopping nature of TSCH, these EB frames are sent on all frequencies.

A mote wishing to join the network listens on some frequency for EBs. It can wait to receive multiple, and can use the join priority in those EBs to identify the best mote to join the network through. Using the ASN and the other timing information of the EB, the new mote synchronizes to the network. Using the slotframe and link information from the EB, it knows how to contact the mote it just joined.

The TSCH standard does not define the steps beyond this "kickstart". These steps can include a security handshake and the addition of more communication links to the new mote's schedule.

#### **A.12. Information Elements**

TSCH introduces the concept of Information Elements (IES). An information element 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), but an unmanaged range is available for extensions.

A 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 motes to exchange information.





### [A.13.](#) Extensibility

The TSCH standard is designed to be extensible. It introduces the mechanisms as "building block" (e.g. links, frame, 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 Payload IEs.

## **Appendix B. TSCH Gotchas**

This section lists features of TSCH which we believe are important and beneficial to the work of 6TSCH.

### **B.1. Collision Free Communication**

TSCH allows you to easily design a schedule which yields collision-free communication. This is done by building the schedule in such a way that at most one link occupies each slotOffset/channelOffset cell. Multiple pairs of neighbor nodes can exchange data at the same time, but on different frequencies. If a deployment is done over a large area, slotOffset/channelOffset cells can be reused by pairs of neighbors sufficiently far apart not to interfere.

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

A TSCH schedule looks like a matrix of width "slotFrame length" and height "number of channels". For a scheduling algorithm, these 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**

In the absence of data traffic, a node is required to synchronize to its time source neighbor(s) periodically not to drift in time. The period at which this needs to happen depends on the stability of the clock source, and on how "early" each node starts listening for data (the "guard time"). Theoretically, with a 10ppm clock and a 1ms guard time, this period can be 100s. Resynchronizing consists in sending any valid frame to the time source neighbor and using the timing information in the ACK to realign the clocks. If this exchange causes the node's radio to be on for 5ms, this yields a radio duty cycle needed to keep synchronized of  $5\text{ms}/100\text{s}=0.005\%$ . While TSCH does require nodes to resynchronize periodically, the cost of doing so is almost negligible.

### **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 than retransmitting on the same frequency. As a result, even when some frequencies are "behaving bad", channel hopping



"smoothen" 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 mote's schedule. It is possible that at some slot, a mote has multiple activities scheduled (e.g. transmit to mote 0xfeed on slotFrame 2, receive from mote 0xbeef 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 mote would transmit to mote 0xfeed on slotFrame 2.



Author's Address

Thomas Watteyne (editor)  
Linear Technology  
30695 Huntwood Avenue  
Hayward, CA 94544  
USA

Phone: +1 (510) 400-2978  
Email: [twatteyne@linear.com](mailto:twatteyne@linear.com)