

6TSCH
Internet-Draft
Intended status: Standards Track
Expires: January 14, 2014

P. Thubert, Ed.
cisco
RA. Assimiti
Nivis
T. Watteyne
Linear Technology / Dust Networks
July 15, 2013

An Architecture for IPv6 over Timeslotted Channel Hopping
draft-thubert-6tsch-architecture-02

Abstract

This document presents an architecture for an IPv6 multilink subnet that is composed of a high speed powered backbone and a number of IEEE802.15.4e TSCH wireless networks attached and synchronized by Backbone Routers. Route Computation may be achieved in a centralized fashion by a Path Computation Element, in a distributed fashion using the Routing Protocol for Low Power and Lossy Networks, or in a mixed mode. The Backbone Routers perform proxy Neighbor discovery operations over the backbone on behalf of the wireless device, so they can share a same subnet and appear to be connected to the same backbone as classical devices.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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 January 14, 2014.

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

1. Introduction	2
2. Terminology	3
3. Applications and Goals	3
4. Overview and Scope	4
5. Centralized vs. Distributed Routing	7
6. Forwarding Models	7
6.1. Track Forwarding	7
6.1.1. Transport Mode	8
6.1.2. Tunnel Mode	8
6.1.3. Tunnel Metadata	9
6.2. Fragment Forwarding	10
6.3. IPv6 Forwarding	11
7. Functional Flows	12
8. Network Synchronization	12
9. TSCH and 6top	12
9.1. 6top	12
9.2. Slotframes and Priorities	13
9.3. Centralized Flow Reservation	13
9.4. Distributed Flow Reservation	13
9.5. Packet Marking and Handling	14
10. Management	14
11. IANA Considerations	14
12. Security Considerations	14
13. Acknowledgements	14
14. References	14
14.1. Normative References	14
14.2. Informative References	15
14.3. External Informative References	16
Authors' Addresses	17

1. Introduction

The emergence of radio technology enabled a large variety of new types of devices to be interconnected, at a very low marginal cost compared to wire, at any range from Near Field to interplanetary distances, and in circumstances where wiring could be less than practical, for instance rotating devices.

At the same time, a new breed of Time Sensitive Networks is being developed to enable traffic that is highly sensitive to jitter and quite sensitive to latency. Such traffic is not limited to voice and video, but also includes command and control operations such as found in industrial automation or in-vehicle sensors and actuators.

At IEEE802.1, the "Audio/Video Task Group", was renamed TSN for Time Sensitive Networking to address Deterministic Ethernet. The IEEE802.15.4 Medium Access Control (MAC) has evolved with IEEE802.15.4e that provides in particular the Timeslotted Channel Hopping (TSCH) mode for industrial-type applications.

Though at a different time scale, both standards provide Deterministic capabilities to the point that a packet that pertains to a certain flow will cross the network from node to node following a very precise schedule, like a train leaves intermediate stations at precise times along its path. The time slotted aspect reduces collisions, and saves energy, and enables to more closely engineer the network for deterministic properties. The channel hopping aspect is a simple and efficient technique to get around statistical interference by WiFi emitters.

This document presents an architecture for an IPv6 multilink subnet that is composed of a high speed powered backbone and a number of IEEE802.15.4e TSCH wireless networks attached and synchronized by backbone routers. Route Computation may be achieved in a centralized fashion by a Path Computation Element (PCE), in a distributed fashion using the Routing Protocol for Low Power and Lossy Networks (RPL), or in a mixed mode. The Backbone Routers perform proxy Ipv6 Neighbor Discovery (ND) operations over the backbone on behalf of the wireless devices, so they can share a same IPv6 subnet and appear to be connected to the same backbone as classical devices.

2. Terminology

The draft uses terminology defined in [I-D.palattella-6tsch-terminology], [I-D.chakrabarti-nordmark-6man-efficient-nd], [RFC5191] and [RFC4080].

It conforms to the terms and models described for IPv6 in [RFC5889] and uses the vocabulary and the concepts defined in [RFC4291] for IPv6.

3. Applications and Goals

The architecture derives from existing industrial standards for Process Control by its focus on Deterministic Networking, in particular with the use of the IEEE802.15.4e TSCH MAC and the centralized path computation element. This approach leverages the TSCH MAC benefits for high reliability against interference, low-power consumption on deterministic traffic, and its Traffic Engineering capabilities. Deterministic Networking applies in particular to open and closed control loops, as well as supervisory control flows, and management.

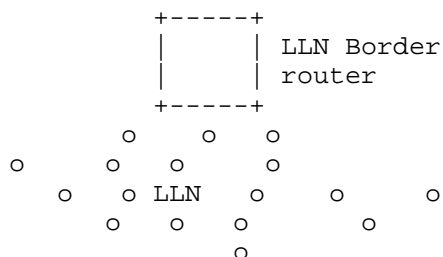
Additional industrial use cases are addressed with the addition of a more autonomic and distributed routing based on RPL. These use cases include plant setup and decommissioning, as well as monitoring of lots of lesser importance measurements such as corrosion and events. RPL also enables mobile use cases such as mobile workers and cranes.

A Backbone Router is included in order to scale the factory plant subnet to address large deployments, with proxy ND and time synchronization over a high speed backbone.

The architecture also applies to building automation that leverage RPL's storing mode to address multipath over a large number of hops, in-vehicule command and control that can be as demanding as industrial applications, commercial automation and asset Tracking with mobile scenarios, home automation and domotics which become more reliable and thus provide a better user experience, and resource management (energy, water, etc.).

4. Overview and Scope

The scope of the present work is a subnet that, in its basic configuration, is made of a IEEE802.15.4e Timeslotted Channel Hopping (TSCH) [I-D.wattheyne-6tsch-tsch-lln-context] MAC Route-Over Low Power Lossy Network (LLN).



The LLN devices communicate over IPv6 [RFC2460] using the 6LoWPAN Header Compression (6LoWPAN HC) [RFC6282]. From the Layer 3 perspective, a single LLN interface (typically an IEEE802.15.4 radio) may be seen as a collection of Links with different capabilities for unicast or multicast services. An IPv6 subnet will span over multiple links, effectively forming a multilink subnet. Within that subnet, Neighbor Devices are discovered with 6LoWPAN Neighbor Discovery (6LoWPAN ND) [RFC6775]. The Routing Protocol for Low Power and Lossy Networks (RPL) [RFC6550] enables routing within the LLN, typically within the multilink subnet in the so called Routing Over fashion. RPL forms Destination Oriented Directed Acyclic Graphs (DODAGs) within instances of the protocol, each instance being associated with an Objective Function (OF) to form a routing topology. A particular LLN device, usually powered, acts as RPL root, 6LoWPAN HC terminator, and LLN Border Router (LBR) to the outside.

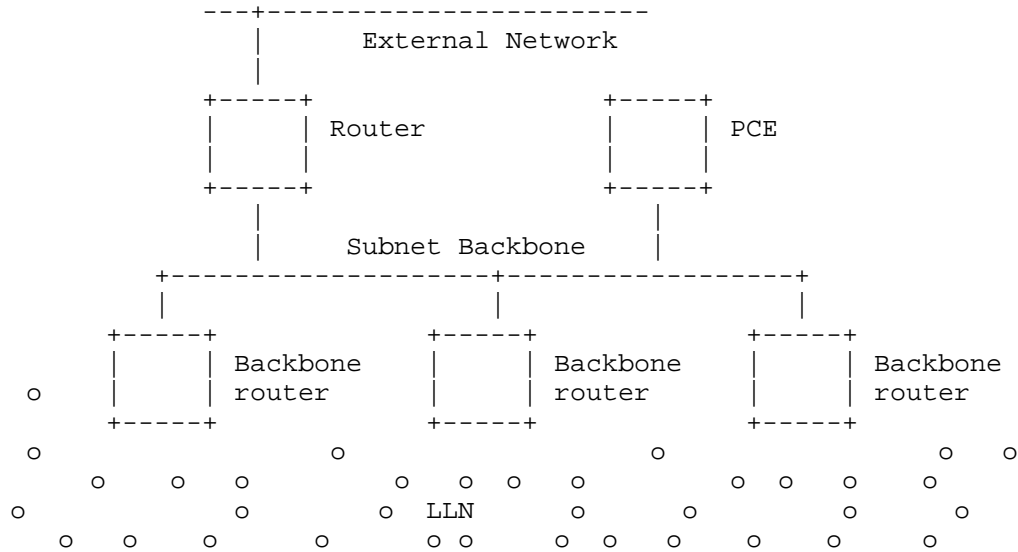
An extended configuration of the subnet comprises multiple LLNs. The LLNs are interconnected and synchronized over a backbone, that can be wired or wireless. The backbone can be a classical IPv6 network, with Neighbor Discovery operating as defined in [RFC4861] and [RFC4862]. The backbone can also support Efficiency aware IPv6 Neighbor Discovery Optimizations [I-D.chakrabarti-nordmark-6man-efficient-nd] in mixed mode as described in [I-D.thubert-6lowpan-backbone-router].

Security is often handled at layer 2 and Layer 4. Authentication during the join process is handled with the Protocol for Carrying Authentication for Network Access (PANA) [RFC5191].

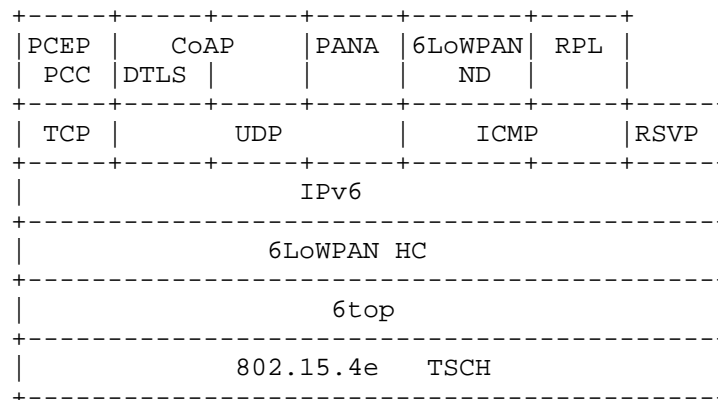
The LLN devices are time-synchronized at MAC level. The MAC coordinator that serves as time source is loosely coupled with the RPL parent; this way, the time synchronization starts at the RPL root and follows the RPL DODAGs with no timing loop.

In the extended configuration, the functionality of the LBR is enhanced to that of Backbone Router (BBR). A BBR is an LBR, but also an Energy Aware Default Router (NEAR) as defined in [I-D.chakrabarti-nordmark-6man-efficient-nd]. The BBR performs ND proxy operations between the registered devices and the classical ND devices that are located over the backbone. 6TSCH BBRs synchronize with one another over the backbone, so as to ensure that the multiple LLNs that form the IPv6 subnet stay tightly synchronized. If the Backbone is Deterministic (such as defined by the Time Sensitive Networking WG at IEEE), then the Backbone Router ensures that the end-to-end deterministic behavior is maintained between the LLN and the

backbone.



The main architectural blocks are arranged as follows:



RPL is the routing protocol of choice for LLNs. (TBD RPL) whether there is a need to define a 6TSCH OF.

(tbd NME) COMAN is working on network Management for LLN. They are considering the Open Mobile Alliance (OMA) Lightweight M2M (LWM2M) Object system. This standard includes DTLS, CoAP (core plus the Block and Observe patterns), SenML and CoAP Resource Directory.

(tbd PCC) need to work with PCE WG to define flows to PCE, and define how to accommodate PCE routes and reservation. Will probably look a lot like GMPLS

(tbd Backbone Router) need to work with 6MAN to define ND proxy. Also need BBR sync sync between deterministic ethernet and 6TSCH

LLNs.

IEEE802.1TSN: external, maintain consistency.

IEEE802.15.4: external, (tbd need updates?).

ISA100.20 Common Network Management: external, maintain consistency.

IoT6 European Project: external, maintain consistency.

5. Centralized vs. Distributed Routing

6TSCH supports a mix model of centralized routes that are computed by a Path Computation Entity and distributed routes that are computed by RPL over a common physical LLN.

Both RPL and the PCE may inject routes in the Routing Tables of the 6TSCH routers. In either case, each route is associated with a topology that is indexed by an instanceID, as defined in RPL [RFC6550]. RPL and PCE rely on shared sources to define Global and Local InstanceIDs.

It is possible for RPL and PCE to share a same topology, in which case the PCE routes have precedence over RPL routes in case of a conflict.

Inside the 6TSCH domain, the flow label is used to indicate the topology that must be used for routing and the associated Routing Tables as discussed in [I-D.thubert-roll-flow-label].

6. Forwarding Models

6TSCH supports three different forwarding model, G-MPLS Track Forwarding (TF), 6LoWPAN Fragment Forwarding (FF) and IPv6 Forwarding (6F).

6.1. Track Forwarding

Track Forwarding is the simplest and fastest. A set of input cells are uniquely bound to a set of output cells, representing a forwarding state that can be used regardless of the upper layer protocol. This model can effectively be seen as a G-MPLS operation in that the information used to switch is not an explicit label but related to other properties of the way the packet was received, a particular cell in the case of 6TSCH. As a result, as long as the TSCH MAC (and Layer 2 security) accepts a frame, that frame can be switched regardless of the protocol, whether this is an IPv6 packet, a 6LoWPAN fragment, or a frame from an alternate protocol such as WirelessHART or ISA100.11a.

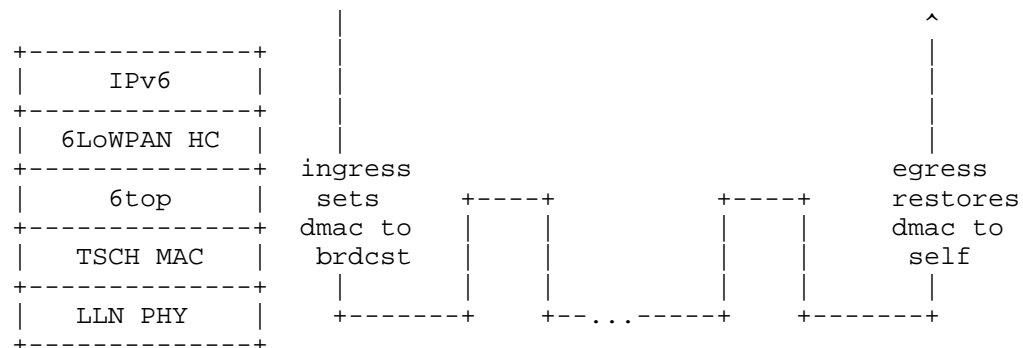
A Track is defined end-to-end as a succession of Timeslots and a Timeslot belongs to at most one Track. For a given iteration of a Slotframe, the Timeslot is associated uniquely with a cell which indicates the channel at which the Timeslot operates for that iteration.

A frame that is forwarded along a Track has a destination MAC address set to broadcast or a multicast address depending on the MAC support. This way, the MAC layer in the intermediate nodes will accept the incoming frame and 6top will switch it without incurring a change in the MAC header. In the case of 802.15.4e, this means effectively broadcast, so that along the Track the short address for the destination is set to 0xFFFF.

Conversely, a frame that is received along a Track with a destination MAC address set to this node is extracted from the Track stream and delivered to the upper layer. A frame with an unrecognized MAC address is just ignored at the MAC layer and thus is not received at the 6top sublayer.

There are 2 modes for a Track, transport mode and tunnel mode.

6.1.1. Transport Mode

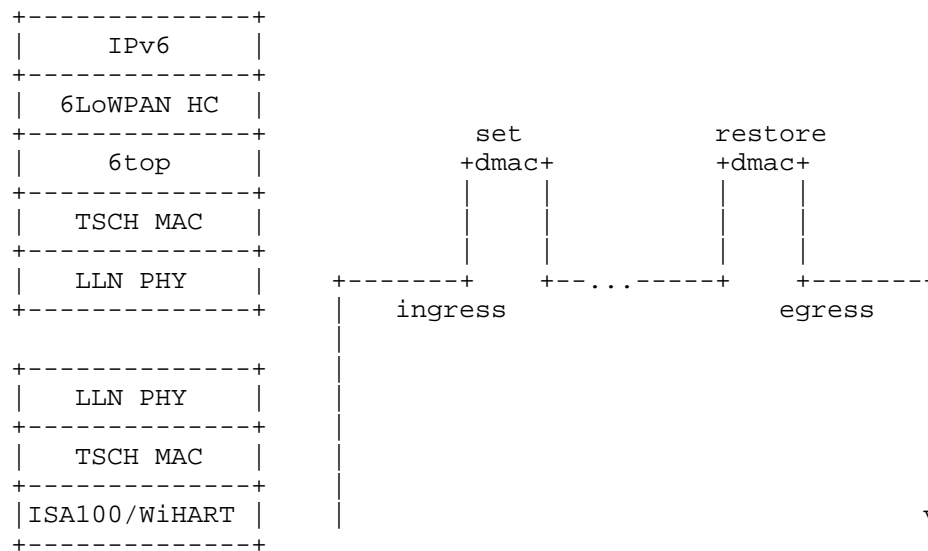


In transport mode, the PDU is associated with flow information that refers uniquely to the Track, so the 6top sublayer can place the frame in the appropriate Timeslot without ambiguity. In the case of IPv6 traffic, the identification of that flow information is transported in the Flow Label in the IPv6 header. Associated with the source IPv6 address, the flow label forms a globally unique identifier for that particular Track that is validated at egress before restoring the dmac and punting to the upper layer.

6.1.2. Tunnel Mode

In tunnel mode, the frames originate from an arbitrary protocol over a compatible MAC that may or may not be perfectly synchronized with the 6TSCH network. An example of this would be a router with a dual radio that is capable to receive and send WirelessHART or ISA100.11a frames with the second radio, by presenting itself as an Access Point or a Backbone Router, respectively.

In that mode, the PCE may coordinate with a WirelessHART Network Manager or an ISA100.11a System Manager in order to specify the flows that are to be transported transparently over the Track.



In that case, the flow information that identifies the Track is uniquely derived from the information at the receiving end, for instance the incoming Timeslots, or an ISA100.11a ContractId. At the ingress 6TSCH router, the packet destination is recognized as self but the flow information indicates that the frame must be tunneled over a particular 6top Track so the packet is not punted to upper layer. Instead, it is passed to the 6top sublayer for switching. The 6top sublayer in the ingress router overrides the destination MAC to broadcast and forwards.

At the egress 6top router, the reverse operation occurs. Based on metadata associated to the Track, the frame is passed to the appropriate link layer with the destination MAC restored.

6.1.3. Tunnel Metadata

Metadata coming with the Track configuration is expected to provide the destination MAC address of the egress endpoint as well as the tunnel mode and specific data depending on the mode, for instance a service access point for frame delivery at egress.

If the tunnel egress point does not have a MAC address that matches the configuration, the Track installation fails.

In transport mode, if the final layer 3 destination is the tunnel termination, then it is possible that the IPv6 address of the destination is compressed at the 6LoWPAN sublayer based on the MAC address. It is thus mandatory at the ingress point to validate that the MAC address that was used at the 6LoWPAN sublayer for compression matches that of the tunnel egress point. For that reason, the node that injects a packet on a Track checks that the destination is effectively that of the tunnel egress point before it overwrites it to broadcast. The 6top sublayer at the tunnel egress point reverts that operation to the MAC address obtained from the tunnel metadata.

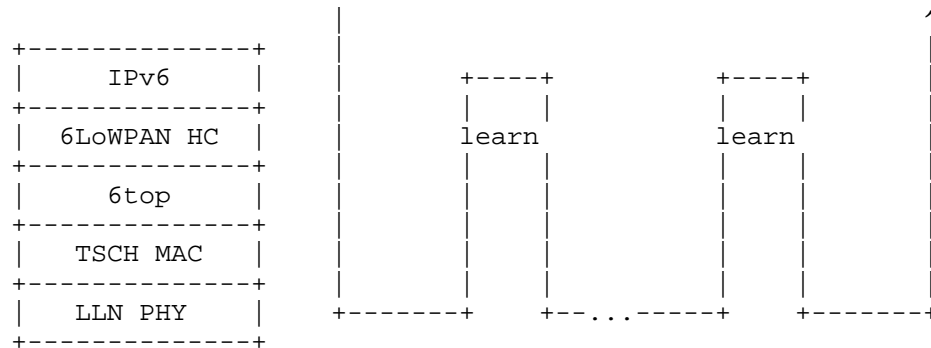
6.2. Fragment Forwarding

Considering that 6LoWPAN packets can be as large as 1280 bytes, which is the IPv6 MTU, and that the non-storing mode of RPL implies Source Routing that requires space for routing headers, and that a 802.15.4 frame with security may carry in the order of 80 bytes of effective payload, an IPv6 packet might be fragmented into more than 16 fragments at the 6LoWPAN sublayer.

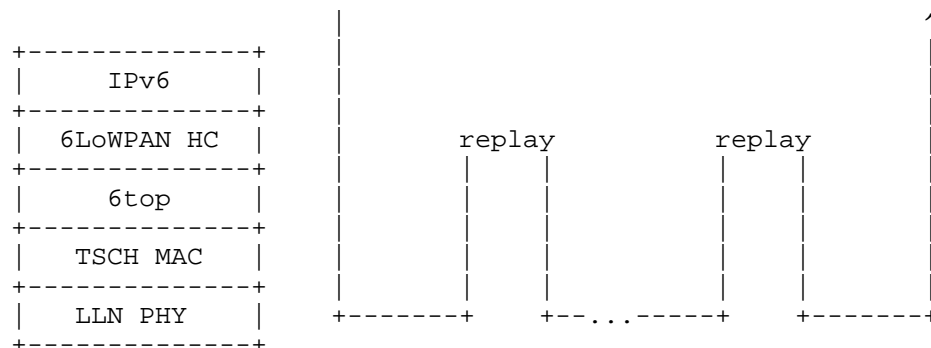
This level of fragmentation is much higher than that traditionally experienced over the Internet with IPv4 fragments, where fragmentation is already known as harmful.

In the case to a multihop route within a 6TSCH network, Hop-by-Hop recomposition would occur at each hop in order to reform the packet and route it. This creates additional latency and forces intermediate nodes to store a portion of a packet for an indetermined time, thus impacting critical resources such as memory and battery.

[I-D.thubert-roll-forwarding-frags] describes a mechanism whereby the datagram tag in the 6LoWPAN Fragment is used as a label for switching at the 6LoWPAN sublayer. The draft allows for a degree of flow control base on an Explicit Congestion Notification, as well as end-to-end individual fragment recovery. In that model, the first fragment is routed based on the IPv6 header that is present in that fragment.



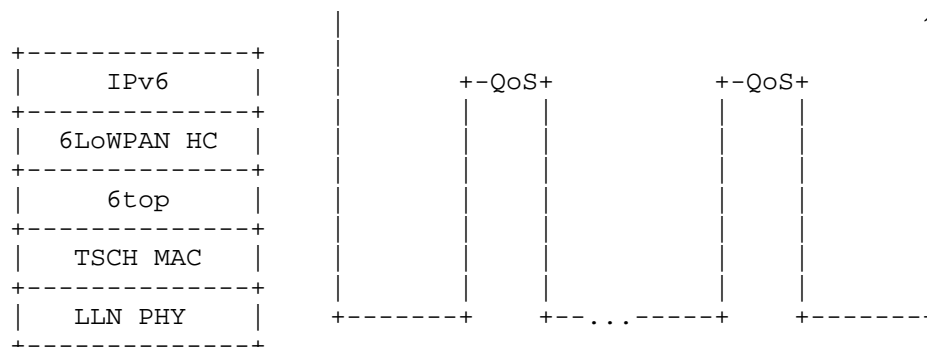
The 6LoWPAN sublayer learns the next hop selection, generates a new datagram tag for transmission to the next hop, and stores that information indexed by the incoming MAC address and datagram tag. The next fragments are then switched based on that stored state.



A bitmap and an ECN echo in the end-to-end acknowledgement enable the source to resend the missing fragments selectively. The first fragment may be resent to carve a new path in case of a path failure. The ECN echo set indicates that the number of outstanding fragments should be reduced.

6.3. IPv6 Forwarding

As the packets are routed at layer 3, traditional QoS and RED operations are expected to prioritize flows with differentiated services. A new class of service for Deterministic Forwarding is being defined to that effect in [I-D.svshah-tsvwg-lln-diffserv-recommendations].



7. Functional Flows

8. Network Synchronization

Nodes in a TSCH are time synchronized. A node keeps synchronized to its time source neighbor(s) through a combination of frame-based and acknowledgment-based synchronization. In order to maximize battery life and network throughput, it is advisable that RPL ICMP discovery and maintenance traffic (governed by the trickle timer) be somehow coordinated with the transmission of time synch packets (especially with enhanced beacons). This could be achieved through an interaction of the 6top sublayer and the RPL objective Function, or could be controlled by the Device Management Entity.

9. TSCH and 6top

9.1. 6top

6top is a sublayer which is the next higher layer to TSCH and which offers a set of commands defining data and management interfaces. 6top is defined in [I-D.draft-wang-6tsch-6top]

The management interface of 6top enables an upper layer to schedule cells and Slotframes in the TSCH schedule.

If the scheduling entity explicitly specifies the slotOffset/channelOffset of the cells to be added/deleted, those cells are marked as "hard". 6top cannot move hard cells in the TSCH schedule. Hard cells are typically used by a central PCE.

6top contains a monitoring process which monitors the performance of cells, and can move a cell in the TSCH schedule when it performs bad. This is only applicable to cells which are marked as "soft". To reserve a soft cell, the higher layer does not indicate the slotOffset/channelOffset of the cell to add, but rather the resulting bandwidth and QoS requirements. When the monitoring process triggers a cell reallocation, the two neighbor nodes communicating over this cell negotiate its new position in the TSCH schedule.

9.2. Slotframes and Priorities

6top uses priority queues to manage concurrent data flows of different priorities. When a packet is received from an higher layer for transmission, the I-MUX module of 6top inserts that packet in the outgoing queue which matches the packet best (DSCP can therefore be used). At each scheduled transmit slot, the MUX module looks for the frame in all the outgoing queues that best matches the cells. If a frame is found, it is given to TSCH for transmission.

9.3. Centralized Flow Reservation

In a centralized setting, a PCE computes the TSCH schedule, and communicates with the different nodes in the network to configure their TSCH schedule. Since it has full knowledge of the network's topology, the PCE can compute a collision-free schedule, which results in a high degree of communication determinism.

The protocol for the PCE to communicate with the nodes is not yet defined. This protocol typically reserves hard cells on the transmitter side of a dedicated cell, and the negotiation protocol of 6top takes care of reserving the same cell on the receiver node.

9.4. Distributed Flow Reservation

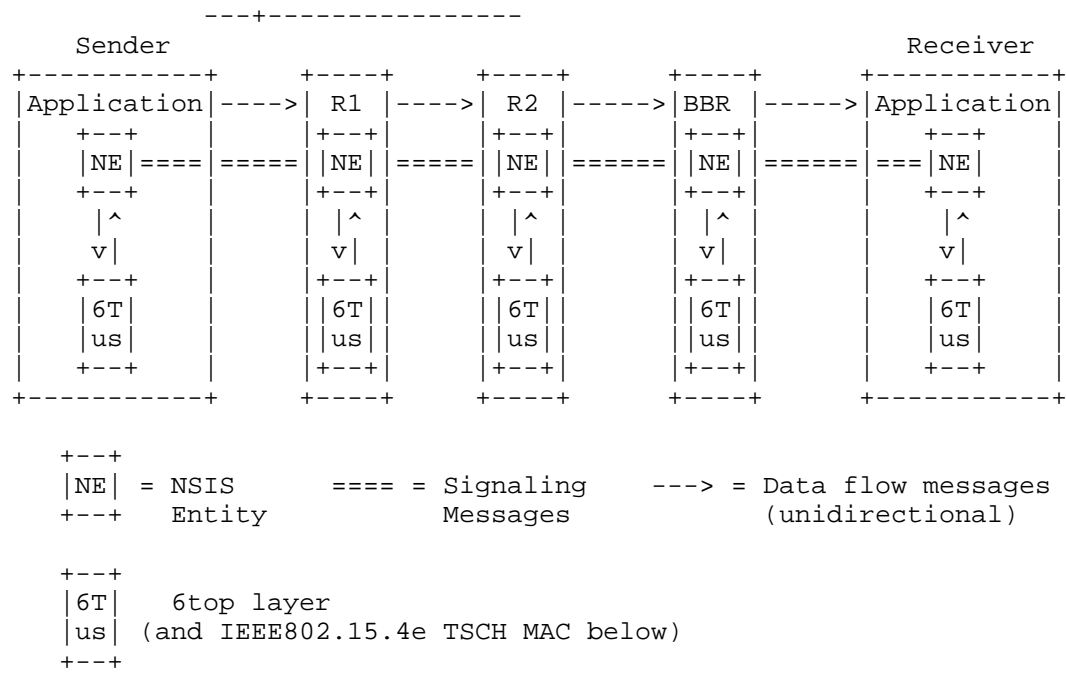
In a distributed setting, no central PCE is present in the network. Nodes use 6top to reserve soft cells with their neighbors. Since no node has full knowledge of the network's topology and the traffic requirements, scheduling collisions are possible, for example because of a hidden terminal problem.

A schedule collision can be detected if two nodes have multiple dedicated cells scheduled to one another. The monitoring process of 6top can be configured to continuously compute the packet delivery ratio of those cells, and it can declare a soft cell to perform bad when the statistics for that cell are significantly worse than for the other cells to the same neighbor.

When this happens, the monitoring process of 6top moves the cell to another location in the 6TSCH schedule, through a re-negotiation procedure with the neighbor.

The entity that builds and maintains the schedule in a distributed fashion is not yet defined.

9.5. Packet Marking and Handling



reservation Deterministic flow allocation (hard reservation of Timeslots) eg centralized RSVP? metrics? Hop-by-hop interaction with 6top. Lazy reservation (use shared slots to transport extra burst and then dynamically (de)allocate) Classical QoS (dynamic based on observation)

10. Management

11. IANA Considerations

This specification does not require IANA action.

12. Security Considerations

This specification is not found to introduce new security threat.

13. Acknowledgements

14. References

14.1. Normative References

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

- [RFC2460] Deering, S.E. and R.M. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC4080] Hancock, R., Karagiannis, G., Loughney, J. and S. Van den Bosch, "Next Steps in Signaling (NSIS): Framework", RFC 4080, June 2005.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W. and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC4862] Thomson, S., Narten, T. and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.
- [RFC5191] Forsberg, D., Ohba, Y., Patil, B., Tschofenig, H. and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA)", RFC 5191, May 2008.
- [RFC5889] Baccelli, E. and M. Townsley, "IP Addressing Model in Ad Hoc Networks", RFC 5889, September 2010.
- [RFC5974] Manner, J., Karagiannis, G. and A. McDonald, "NSIS Signaling Layer Protocol (NSLP) for Quality-of-Service Signaling", RFC 5974, October 2010.
- [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.
- [RFC6775] 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.

14.2. Informative References

- [I-D.chakrabarti-nordmark-6man-efficient-nd]
Chakrabarti, S., Nordmark, E. and M. Wasserman,
"Efficiency aware IPv6 Neighbor Discovery Optimizations",
Internet-Draft draft-chakrabarti-nordmark-6man-efficient-
nd-01, November 2012.
- [I-D.draft-wang-6tsch-6top]

Wang, Q., Ed., Vilajosana, X. and T. Watteyne, "6TSCH Operation Sublayer (6top). draft-wang-6tsch-6top-00 (work in progress) ", July 2013.

[I-D.ohba-6tsch-security]

Chasko, S., Das, S., Lopez, R., Ohba, Y., Thubert, P. and A. Yegin, "Security Framework and Key Management Protocol Requirements for 6TSCH", Internet-Draft draft-ohba-6tsch-security-01, July 2013.

[I-D.palattella-6tsch-terminology]

Palattella, M., Thubert, P., Watteyne, T. and Q. Wang, "Terminology in IPv6 over Time Slotted Channel Hopping", Internet-Draft draft-palattella-6tsch-terminology-00, March 2013.

[I-D.svshah-tsvwg-lln-diffserv-recommendations]

Shah, S. and P. Thubert, "Differentiated Service Class Recommendations for LLN Traffic", Internet-Draft draft-svshah-tsvwg-lln-diffserv-recommendations-00, February 2013.

[I-D.svshah-tsvwg-lln-diffserv-recommendations]

Shah, S. and P. Thubert, "Differentiated Service Class Recommendations for LLN Traffic", Internet-Draft draft-svshah-tsvwg-lln-diffserv-recommendations-00, February 2013.

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

Thubert, P., "6LoWPAN Backbone Router", Internet-Draft draft-thubert-6lowpan-backbone-router-03, February 2013.

[I-D.thubert-roll-flow-label]

Thubert, P., "Use of the IPv6 Flow Label within an LLN", Internet-Draft draft-thubert-roll-flow-label-02, November 2012.

[I-D.thubert-roll-forwarding-frags]

Thubert, P. and J. Hui, "LLN Fragment Forwarding and Recovery", Internet-Draft draft-thubert-roll-forwarding-frags-01, February 2013.

[I-D.vilajosana-6tsch-basic]

Vilajosana, X. and K. Pister, "Minimal 6TSCH Configuration", Internet-Draft draft-vilajosana-6tsch-basic-01, July 2013.

[I-D.watteyne-6tsch-tsch-lln-context]

Watteyne, T., "Using IEEE802.15.4e TSCH in an LLN context: Overview, Problem Statement and Goals", Internet-Draft draft-watteyne-6tsch-tsch-lln-context-01, February 2013.

14.3. External Informative References

- [HART] www.hartcomm.org, "Highway Addressable Remote Transducer, a group of specifications for industrial process and control devices administered by the HART Foundation", .
- [IEEE802.1TSNTG] IEEE Standards Association, "IEEE 802.1 Time-Sensitive Networks Task Group", March 2013, <<http://www.ieee802.org/1/pages/avbridges.html>>.
- [ISA100.11a] ISA, "ISA100, Wireless Systems for Automation", May 2008, <<http://www.isa.org/Community/SP100WirelessSystemsforAutomation>>.

Authors' Addresses

Pascal Thubert, editor
Cisco Systems, Inc
Building D
45 Allee des Ormes - BP1200
MOUGINS - Sophia Antipolis, 06254
FRANCE

Phone: +33 497 23 26 34
Email: pthubert@cisco.com

Robert Assimiti
Nivis
1000 Circle 75 Parkway SE, Ste 300
Atlanta, GA 30339
USA

Phone: +1 678 202 6859
Email: robert.assimiti@nivis.com

Thomas Watteyne
Linear Technology / Dust Networks
30695 Huntwood Avenue
Hayward, CA 94544
USA

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

6TSCH
Internet-Draft
Intended status: Informational
Expires: November 24, 2013

T. Watteyne, Ed.
Linear Technology
MR. Palattella
University of Luxembourg
LA. Grieco
Politecnico di Bari
May 23, 2013

Using IEEE802.15.4e TSCH in an LLN context:
Overview, Problem Statement and Goals
draft-watteyne-6tsch-tsch-lln-context-02

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 November 24, 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

1. Introduction	2
2. TSCH in the LLN Context	4
3. Problems and Goals	6
3.1. Network Formation	7
3.2. Network Maintenance	7
3.3. Multi-Hop Topology	8
3.4. Routing and Timing Parents	8
3.5. Resource Management	8
3.6. Dataflow Control	9
3.7. Deterministic Behavior	9
3.8. Path Computation Engine	9
3.9. Secure Communication	9
4. Acknowledgements	10
5. References	10
5.1. Normative References	10
5.2. Informative References	10
5.3. External Informative References	13
Appendix A. TSCH Protocol Highlights	15
A.1. Timeslots	15
A.2. Slotframes	16
A.3. Node TSCH Schedule	16
A.4. Cells and Bundles	16
A.5. Dedicated vs. Shared Cells	17
A.6. Absolute Slot Number	17
A.7. Channel Hopping	17
A.8. Time Synchronization	18
A.9. Power Consumption	19
A.10. Network TSCH Schedule	19
A.11. Join Process	20
A.12. Information Elements	20
A.13. Extensibility	21
Appendix B. TSCH Gotchas	21
B.1. Collision Free Communication	21
B.2. Multi-Channel vs. Channel Hopping	21
B.3. Cost of (continuous) Synchronization	21
B.4. Topology Stability	22
B.5. Multiple Concurrent Slotframes	22
Authors' Addresses	22

1. Introduction

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

In particular, this document describes the main issues arising from the adoption of the IEEE802.15.4e TSCH in the LLN context, mainly following the terminology defined in [I-D.palattella-6tsch-terminology].

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 nodes consume 10's of micro-amps on average [CurrentCalculator] with end-to-end packet delivery ratios over 99.999% [doherty07channel].

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 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].

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-configure IP addresses, or manage keying material.

In other words, IEEE802.15.4e 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

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 started working in 2007 on specifications for transmitting IPv6 packets 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 is too large to fit in an IEEE802.15.4 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 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]. 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 destination, building a gradient. 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 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. RPL therefore 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 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 is different for traditional 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 that schedule which 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 will refer to this entity by the generic name "6TSCH", without loss of generality. In particular, we

do not assume any specific nature of 6TSCH, whether an adaptation layer, a distributed reservation protocol, a centralized path computation engine, or any combination thereof.

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

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 the 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 through which to join the network.
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.
4. Define a mechanism to secure the joining process and the subsequent optional process of scheduling more communication links.

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 Enhanced Beacons 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 to gather 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 routing 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 slotframe.
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 to stop accepting packets. 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 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 links.

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. 6TSCH need to:

1. Ensure timely delivery of such data.
2. Provide a mechanism for such deterministic flows to coexist with bursty or infrequent traffic flows of different priorities.

3.8. Path Computation Engine

As highlighted in [I-D.phinney-roll-rpl-industrial-applicability], bandwidth allocation and multi-hop routes can be optimized by an external Path Computation Engine (PCE). 6TSCH need to:

1. Provide a mechanism for an external PCE to be able to control the entire schedule of the network, including the slotframes, links and time source neighbor assignment.
2. Define a optional mechanism for the schedule managed by this PCE to coexist with scheduling elements (slotframes, links) managed up by a different mechanism such as a distribute scheduling algorithm.

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. 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. Acknowledgements

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

5. References

5.1. Normative References

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

5.2. 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-Namespace 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-01 (work in progress), February 2013.
- [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-12 (work in progress), March 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-17 (work in progress), March 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-04 (work in progress), February 2013.

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

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

[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-02 (work in progress), February 2013.

[I-D.ietf-core-coap]

Shelby, Z., Hartke, K., and C. Bormann, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-16 (work in progress), May 2013.

[I-D.palattella-6tsch-terminology]

Palattella, M., 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.

5.3. 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, "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>.
- [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>>.

[tinkal0decentralized] 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>.

[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>.

[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, <<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 nodes in a TSCH network are synchronized. Time is sliced up into timeslots. A timeslot 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 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 a 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. Node TSCH Schedule

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

Once a mote obtains its schedule, it executes it:

- o For each transmit cell, the mote checks whether there is a packet in the outgoing buffer which matches the neighbor written in the schedule information for that timeslot. If there is none, the mote keeps its radio off for the duration of the timeslot. 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 transmitting.
- o For each receive cell, 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 can send back an 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. Cells and Bundles

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 element of the schedule 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) within a given slotframe, is called a "scheduled cell".

If there is a lot of data flowing from mote A to mote B, the schedule might contain multiple cells from A to B, at different times. Multiple cells scheduled to the same neighbor are typically equivalent, i.e. the MAC layer sends the packet on whichever of these cells happens to show 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 mote A to transmit to mote B. IEEE802.15.4e allows also to mark a cell as shared. In a shared cell, multiple motes can transmit at the same time, on the same frequency. To avoid contention, TSCH defines a back-off algorithm for shared cells.

A scheduled cell 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 cell 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 k is the slotframe cycle (i.e., the number of slotframe occurrences over time), S the slotframe size and t the slotOffset. 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 used by CCM*.

A.7. Channel Hopping

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

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

The function F consists of a look-up table containing the set of available channels. The value nFreq (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 scheduled cell, 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 channeloffsets, nFreq , 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 frequency "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.

In detail, each mote periodically synchronizes its network clock to at least one other mote, and it also provides its network time to its neighbors. It is up to the entity that manages the schedule to assign adequate time source neighbor(s) to each mote, i.e., to indicate in the schedule which of its neighbor(s) are its "time source neighbors". While setting the time source neighbor, it is

important to avoid synchronization loops, which could result in the formation of independent clusters of motes.

Typically, in a IEEE802.15.4e TSCH network, time propagates outwards from the PAN coordinator (i.e., the root node). But the direction of time propagation is independent of data flow in the network. A new mote joining a TSCH network, because it does not have a schedule yet, maintains time synchronization, using the information carried by the Enhanced Beacons (EBs), sent by the advertising motes.

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. In detail, in the IEEE 802.15.4e standard two methods are defined 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 mote in its acknowledgment. Thus, in this case, it is the sender mote that synchronizes to the clock of the receiver. In Frame-Based synchronization, the receiver uses the computed delta for adjusting its own clock. Therefore, it is the receiver mote that synchronizes to the clock of the sender.

Different synchronization policies are possible. Motes can keep synchronization exclusively by exchanging EBs. Motes can also keep synchronized by periodically sending valid frames to time source neighbors to use the acknowledgement to resynchronize. Both method (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 mote can perform during a timeslot: transmit, receive, or sleep. Each of these operations has some energy cost associated to them, the exact value depending on the hardware used. Given the schedule of a mote, it is straightforward to calculate the expected average power consumption of that mote.

A.10. Network TSCH Schedule

The schedule defines entirely the synchronization and communication between motes. By adding/removing cells 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 energy 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 cells 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 through which to join the network. 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 scheduled cells 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), 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 motes to exchange information.

A.13. Extensibility

The TSCH standard is designed to be extensible. It introduces the mechanisms as "building block" (e.g., cells, boundles, 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 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 one to easily design a schedule which yields collision-free communication. This is done by building the schedule with dedicated cells in such a way that at most one node can communicate 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. 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 size", S, and of height "number of frequencies", nFreq. 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

When there is traffic in the network, nodes which are communicating implicitly re-synchronize 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 30s), nodes can exchange an empty data frame, often referred to as a "Keep-alive" message, to re-synchronize. The frequency at which such message 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 10ppm clock and a 1ms guard time, this period can be 100s. When acknowledgment-based synchronization is

used, re-synchronizing consists in sending any valid frame to the time source neighbor and using the timing information in the ACK to realign the clocks. Assuming this exchange causes the mote'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 requires motes to resynchronize periodically, the cost of doing so can be considered almost negligible in many applications.

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 motes differently on each frequency. If a transmission from mote A to mote 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 "smoothes" 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 timeslot, a mote has multiple activities scheduled (e.g. transmit to mote B on slotframe 2, receive from mote 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 mote would transmit to mote B on slotframe 2.

Authors' Addresses

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

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: 00390805963911
Email: a.grieco@poliba.it