

6TiSCH  
Internet-Draft  
Intended status: Informational  
Expires: January 5, 2015

D. Dujovne, Ed.  
Universidad Diego Portales  
LA. Grieco  
Politecnico di Bari  
MR. Palattella  
University of Luxembourg  
N. Accettura  
University of California Berkeley  
July 4, 2014

6TiSCH On-the-Fly Scheduling  
draft-dujovne-6tisch-on-the-fly-03

Abstract

This document describes the environment, problem statement, and goals of On-The-Fly (OTF) scheduling, a Layer-3 mechanism for 6TiSCH networks. The purpose of OTF is to dynamically adapt the aggregate bandwidth, i.e., the number of reserved softcells between neighbor nodes, based on the specific application constraints to be satisfied. When using OTF, softcell and bundle reservation is distributed: through the 6top interface, neighbor nodes negotiate the cell(s) to be (re)allocated/deleted, with no intervention needed of a centralized entity. This document aims at defining a module which uses the functionalities provided by the 6top sublayer to (i) extract statistics and (ii) determine when to reserve/delete soft cells in the schedule. The exact reservation and deletion algorithm, and the number and type of statistics to be used in the algorithm are out of scope. OTF deals only with the number of softcells to be reserved/deleted; it is up to 6top to select the specific soft cells within the TSCH schedule.

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 5, 2015.

#### Copyright Notice

Copyright (c) 2014 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. Allocation policy . . . . .	3
3. Allocation methods . . . . .	5
4. Cell and Bundle Reservation/Deletion . . . . .	6
5. Getting statistics and other information about cells through 6top . . . . .	7
6. Events triggering algorithms in OTF . . . . .	8
7. Bandwidth Estimation Algorithms . . . . .	9
8. OTF external CoAP interface . . . . .	10
9. Acknowledgments . . . . .	11
10. References . . . . .	11
10.1. Informative References . . . . .	11
10.2. External Informative References . . . . .	12
Authors' Addresses . . . . .	12

#### 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.

On-The-Fly (OTF) scheduling is a 1-hop protocol with which a node negotiates the number of soft cells scheduled with its neighbors, without requiring any intervention of a centralized entity (e.g., a PCE). This document describes the OTF allocation policies and methods used by two neighbors to allocate one or more softcells in a distribution fashion. It also proposes an algorithms for estimating the required bandwidth (BW). This document defines the interface

between OTF and the 6top sublayer ([I-D.wang-6tisch-6top]), to collect and retrieve statistics, or allocate/delete cells and bundles. This document defines a framework; the algorithm and statistics used are out of scope. This draft follows the terminology defined in [I-D.ietf-6tisch-terminology] and addresses the open issue related to the scheduling mechanisms raised in [I-D.ietf-6tisch-tsch].

## 2. Allocation policy

OTF is a distributed scheduling protocol which increases/decreases the bandwidth between two neighbor nodes (i.e., adding/deleting softcells) by interacting with the 6top sublayer. It retrieves statistics from 6top, and uses that information to trigger 6top to add/delete softcells to a particular neighbor. The algorithm which decides when to add/delete softcells is out of scope. For example, 6top might decide to add a cell if some queue of outbound frames is overflowing. Similarly, OTF can delete cells when the queue has been empty for some time. OTF only triggers 6top to add/delete the soft cells, it is the responsibility of the 6top sublayer to determine the exact slotOffset/channelOffset of those cells. In this document, the term "cell" and "soft cell" are used interchangeably.

All the softcells allocated are part of best effort track, i.e. with TrackID=00, as defined in [I-D.wang-6tisch-6top]. These cells can be used for forwarding any packet in the queue, regardless of the specific track it belongs to. OTF manages the global bandwidth requirements between two neighbor nodes; per-track management is currently out of scope.

OTF is prone to schedule collisions. Nodes might not be aware of the cells allocated by other pairs of nodes. A schedule collision occurs when the same cell is allocated by different pairs in the same interference space. The probability of having allocation collision may be kept low by grouping cells into chunks (see [I-D.ietf-6tisch-terminology] and [I-D.ietf-6tisch-architecture] for more details). The use of chunks is outside the scope of this current version of the OTF draft.

The "allocation policy" is the algorithm used by OTF to decide when to increase/decrease the bandwidth allocated between two neighbor nodes in order to satisfy the traffic requirements. These requirements can be expressed in terms of throughput, latency or other constraints.

An OTF allocation policy MAY be defined according to a combination of two different approaches: reactive and proactive.

In a reactive approach, the allocation policy follows the increased/decreased need for bandwidth. Upon reception of a bandwidth request, OTF sends softcell allocation requests to the 6top sublayer. OTF estimates the number of cells to be allocated per neighbor. If the traffic exchanged between two neighbors reduces, OTF asks 6top to de-allocate one or more cells. Once the cells are deleted, 6top notifies OTF, which updates its internal state.

In a proactive approach, the allocation policy over-provisions the number of cells reserved in a bundle between two neighbors, i.e., cells are scheduled in advance. When OTF issues a bundle allocation request to 6top, it indicates the desired size of the bundle and the TrackID=00. 6top selects the cells belonging to the bundle on the best effort track. Based on the network traffic conditions (e.g., queue utilization), some portion of those cells are used for communication. In any case, allocated cells within a bundle are consecutive, starting from the first cell in the block. The cells which are not currently used, are still reserved for that pair of nodes, for possible future use.

It is up to the implementor to select the approach most appropriate for the application. The reactive approach is, in general, be more energy-efficient (it allocates only the cells needed), at the expense of increased cell allocation latency (negotiating to add/delete cells takes some time).

The proactive approach compared to the reactive one reduces the cell allocation latency. Cells within a bundle are over-provisioned, and a priori scheduled. When needed, the 6top sublayer of the node can allocate them, without going through any negotiation phase with the 6top layer of the neighbor node. Thus, the proactive approach provides a low-delay response after a surge in bandwidth usage. In fact, soft cells within a bundle are already scheduled and become immediately available, upon bandwidth request, without the need of a negotiation phase. The use of bundles does force the receiver module of the node to be active during the whole length of the bundle, resulting in increased power consumption.

This document introduces the following parameters to accomplish both approaches previously described:

**SCHEDULEBW:** The amount of cells scheduled in a bundle on the best effort track between two neighbors.

**REQUIREDBW:** Bandwidth requested by OTF to 6top, a non-negative number. How this is computed is out of the scope. It MAY be the an instantaneous bandwidth request, or a value averaged on several measurement, or an over-provisioned value.

PROACTIVETHRESH: Threshold parameter to introduce pro-activity in the allocation policy, described below. It is a non-negative bandwidth value. What value to use is application specific and out of scope. The maximum acceptable value for this parameter is equal to the current SCHEDULEBW.

The OTF allocation policy compares the required bandwidth against the scheduled one, using the pro-activity threshold for bounding the signaling overhead due to negotiations of new cells. In details:

1. If REQUIREDDBW is greater than SCHEDULEBW, OTF asks 6top to add REQUIREDDBW-SCHEDULEBW cells to the bundle on the best effort track.
2. If REQUIREDDBW is greater or equal than SCHEDULEBW-PROACTIVETHRESH, and it is lower than or equal to SCHEDULEBW, OTF does not perform any bundle resizing, since the scheduled bandwidth is sufficient for managing the current traffic conditions.
3. If REQUIREDDBW is lower than SCHEDULEBW-PROACTIVETHRESH, OTF asks 6top to delete SCHEDULEBW-PROACTIVETHRESH-REQUIREDDBW from the bundle on the best effort track.

A purely reactive approach uses PROACTIVETHRESH=0. In this case, OTF does not perform any allocating/deallocating operation when the required bandwidth is equal to the scheduled one.

A purely proactive approach uses PROACTIVETHRESH=SCHEDULEBW. In this case, OTF resizes the bundle only when the required bandwidth is greater than the scheduled one.

### 3. Allocation methods

Beyond the allocation policies that describe the approach used by OTF for fulfilling the node bandwidth requests, the OTF framework also includes Allocation Methods that specify how OTF issues commands to the 6top sublayer. In other words, the allocation methods represent the mechanisms that are used by the allocation policies.

In detail, OTF includes two distinct allocation methods: soft cell and bundle allocation methods. Each Allocation Policy can use either one or both allocation methods. As specified in [I-D.wang-6tisch-6top], 6top provides a set of commands that allows OTF to allocate/delete soft cells. The same set of commands can be used for reserving bundles.

With the soft cell allocation method, OTF has 6top reserve a single soft cell on the best effort track, for allowing a given node to exchange traffic with a specific neighbor. The 6top layer allocates and maintains this cell. If a bundle is already reserved between the same pair of neighbors, on the same track, this request translates into a bundle resize request. The newly allocated cell increase the size of the already existing bundle. Similarly, when OTF realizes there is a reduction of traffic exchanged between the two neighbors, it may asks 6top to delete a softcell from the best effort track, i.e. to decrease the size of the bundle on the best effort track. If no bundle with TrackID=00 exist, the 6top softcell create command generates a new bundle of size 1.

With the bundle allocation method, OTF sends bundle allocation requests to 6top sublayer, specifying the bundle size (the number of soft cells) and the TrackID=00. Scheduling N softcells is equivalent to asking for a bundle of size N. The cells within the bundle are allocated by 6top (and thus, used for traffic exchange) only afterwards, according to the nodes bandwidth need.

#### 4. Cell and Bundle Reservation/Deletion

In order to reserve/delete softcells, OTF interacts with 6top sublayer. To this aim OTF uses the following set of commands offered by 6top: CREATE.softcell, and DELETE.softcell. When creating (deleting) a softcell, OTF specified the track the cell belongs to (i.e., best effort track, TrackID=00), but not its slotOffset and channelOffset. If at least one cell on the best effort track already exists, the CREATE.softcell and DELETE.softcell, translate into INCREASE and DECREASE the bundle size, respectively. 6top is responsible for picking the specific cell to be added/deleted within the bundle. Before being able to do so, source and destination nodes go through a cell negotiation process. This process is out of scope of 6top and OTF. In order to reserve a best effort bundle, OTF uses the CREATE.softcell command, set TrackID=00, but asks 6top for multiple softcells. Following OTF request, 6top either (i) creates a new bundle, if no cells were reserved already on the best effort track, or (ii) increases the bundle size of the already existing best-effort bundle. By using the DELETE.softcell command, and asking for deleting multiple softcells, OTF has 6top delete the entire best effort bundle.

OTF provides a policy for 6top to generate CREATE/DELETE.softcells commands, policy that is out of 6top scope [I-D.wang-6tisch-6top]. Such policy is not the only one that can be used by 6top. Others may be defined in the future.

## 5. Getting statistics and other information about cells through 6top

Statistics are kept in 4 data structure of 6top MIB: CellList, MonitoringStatusList, NeighborList, and QueueList.

CellList provides per-cell statistics. From this list, an upper layer can get per-bundle statistics. OTF may have access to the CellList, by using the CoAP-YANG Model, but actually cell-specific statistics are not significant to OTF, since softcells can be re-allocated in time by 6top itself, based on network conditions.

MonitoringStatusList provides per-neighbor and slotframe statistics. From it an upper layer (e.g., OTF) can get per bundle overview of scheduling and its performance. Such list contains information about the number of hard and soft cells reserved to a given node with a specific neighbor, and the QoS (that can be expressed in form of different metrics: PDR, ETX, RSSI, LQI) on the actual bandwidth, and the over-provisioned bandwidth (which includes the over-provisioned cells). 6top can use such list to operate 6top Monitoring Functions, such as re-allocating cells (by changing their slotOffset and/or channelOffset) when it finds out that the link quality of some softcell is much lower than average. Unlike 6top, OTF does not operate any re-allocation of cells. In fact, OTF can ask for more/less bandwidth, but cannot move any cell within the schedule. Thus, the 6top Monitoring function is useful to OTF, because it can provide better cells for a given bandwidth requirement, specified by OTF. For instance, OTF may require some additional bandwidth (e.g. 2 cells in a specific slotframe) with PDR = 75%; then, 6top will reserve 3 slots in the slotframe to meet the bandwidth requirement. In addition, when the link quality drop to 50%, 6top will reserve 4 slots to keep meeting the bandwidth requirement. Given that OTF operates on the global bandwidth between two neighbor nodes, it does not need to be informed from 6top about cells' re-allocation.

NeighborList provides per-neighbor statistics. From it, an upper layer can understand the connectivity of a pair of nodes. Based on the quality of the link, e.g., LQI under threshold, OTF may ask 6top to delete some cells, in order to reserve them for better-connected links.

QueueList provides per-Queue statistics. From it, an upper layer can know the traffic load. OTF, based on such queue statistics (e.g., average length of the queue, average age of the packet in queue, etc.) may trigger a 6top CREATE.softcell (DELETESoftcell) command for increasing (decreasing) the bandwidth and be able to better serve the packets in the queue.

## 6. Events triggering algorithms in OTF

The Algorithms running within OTF MUST be event-oriented. As a consequence, OTF requires to connect the algorithms with external events to trigger their execution. The algorithm also generates one or more events when it is executed, such as a new softcell allocation. Both type of events, the one which trigger the algorithm and the ones which are generated by the execution of the algorithm are called OTF events.

The following notation is used on the definition of OTF events:

$BW \leftarrow BWA(B,T,S(T))$  where:

BWA: Bandwidth allocation algorithm

BW: Bandwidth

T: Best Effort Track

B: Bundle

$S(B,T)$  Statistics for bundle B on track T

$M(B,T)$ : Actual bundle size for bundle B on track T

The OTF events are defined as:

Event A: A new bundle B on track T is created. The OTF events generated by the algorithm are:

1. Add a new entry in the storage M for bundle B on track T.
2. Ask 6top for  $S(B,T)$ .
3.  $BW \leftarrow BWA(B,T,S(T))$ .
4. Ask 6top to allocate a bundle of size BW.
5.  $M(B,T) \leftarrow BW$ .

Event B: A packet is waiting to be transmitted on any track, but no cell is available (i.e., saturation). The OTF events generated by the algorithm are:

1. Collect stats S from 6top.
2.  $BW \leftarrow BWA(B,T,S(T))$ .



3. Ask 6top to increase the bundle size up to BW.
4. If (allocation successful) then  $M(B,T) \leftarrow BW$ .

Event C: The usage of a bundle B on track T is too low, below a pre-established threshold. The OTF events generated by the algorithm are:

1. Collect stats S from 6top.
2.  $BW \leftarrow BWA(B,T,S(T))$ .
3. Ask 6top to decrease the bundle size to BW.
4. If (allocation successful) then  $M(B,T) \leftarrow BW$ .

Event D: The usage of a bundle B on track T is too high, above a pre-established threshold. The OTF events generated by the algorithm are:

1. Collect stats S from 6top.
2.  $BW \leftarrow BWA(B,T,S(T))$ .
3. Ask 6top to increase the bundle size to BW.
4. If (allocation successful) then  $M(B,T) \leftarrow BW$ .

Event E: Bundle B on track T is deleted. The OTF events generated by the algorithm are:

1. purge  $M(B,T)$ .

## 7. Bandwidth Estimation Algorithms

OTF supports different bandwidth estimation algorithms that can be used by a node in a 6TiSCH network for checking the current traffic condition and thus the actual bandwidth usage. By doing so, one can adapt (increase or decrease) the number of scheduled cells/bundles for a given pair of neighbors (e.g., parent node and its child), according to their needs. OTF supports several bandwidth estimation algorithms numbered 0 to 255 in the OTF implementation. The first algorithm (0) is reserved to the default algorithm that is described below. By using SET and GET commands, one can set the specific algorithm to be used, and get information about which algorithm is implemented.

Steps of the default bandwidth estimation algorithm, running over a parent node:

- Step 1: Collect the bandwidth requests from child nodes (incoming traffic).
- Step 2: Collect the node bandwidth requirement from the application (self/local traffic).
- Step 3: Collect the current outgoing scheduled bandwidth (outgoing traffic).
- Step 4: If (outgoing < incoming + self) then SCHEDULE soft cells/bundles to satisfy bandwidth requirements.
- Step 5: If (outgoing > incoming + self) then DELETE the soft cells that are not used.
- Step 6: Return to step 1.

The default bandwidth estimation algorithm introduced in this document adopts a reactive allocation policy; it is possible to configure proactivity by using a given PROACTIVETHRESH value. In this case, at Step 4, new soft cells will be scheduled, using the cell allocation method, only if there are no free cells in the bundle that can satisfied the current bandwidth request. The node asks 6top for increasing the bundle size by using the bundle allocation method.

## 8. OTF external CoAP interface

In order to select the current OTF algorithm and provide functional parameters from outside OTF, this module uses CoAP with YANG as the data model. The algorithm number and the parameters MUST be invoked in different CoAP calls.

The path to select the algorithm is '6t/e/otf/alg' with A as the algorithm number.

```

Header  +-----+
        | POST                                     |
        +-----+
Uri-Path| /6t/e/otf/alg                           |
        +-----+
Options | CBOR( {AlgNo: 123} )                     |
        +-----+

```

Figure 1: Algorithm number POST message

To obtain the current algorithm number:

```

+-----+
Header | GET |
+-----+
Uri-Path | /6t/e/otf/alg |
+-----+
Options | Accept: application/cbor |
+-----+

```

Figure 2: Algorithm number GET message

An example is: 'coap://[aaaa::1]/6t/e/otf/alg'

The current algorithm parameter path is '6t/e/otf/alg/par'.

```

+-----+
Header | POST |
+-----+
Uri-Path | /6t/e/otf/alg/par |
+-----+
Options | CBOR( {Par: 0x1234} ) |
+-----+

```

Figure 3: Algorithm number POST message

An example follows: 'coap://[aaaa::1]/6t/e/otf/alg/par'

## 9. Acknowledgments

Special thanks to Prof. Kris Pister for his valuable contribution in designing the default Bandwidth Estimation Algorithm, and to Prof. Qin Wang for her support in defining the interaction between OTF and 6top sublayer.

Thanks to the Fondecyt 1121475 Project, to INRIA Chile "Network Design" group and to the IoT6 European Project (STREP) of the 7th Framework Program (Grant 288445).

## 10. References

### 10.1. Informative References

- [I-D.ietf-6tisch-terminology]  
 Palattella, M., Thubert, P., Watteyne, T., and Q. Wang,  
 "Terminology in IPv6 over the TSCH mode of IEEE  
 802.15.4e", draft-ietf-6tisch-terminology-01 (work in  
 progress), February 2014.

[I-D.ietf-6tisch-architecture]

Thubert, P., Watteyne, T., and R. Assimiti, "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-architecture-02 (work in progress), June 2014.

[I-D.ietf-6tisch-tsch]

Watteyne, T., Palattella, M., and L. Grieco, "Using IEEE802.15.4e TSCH in an LLN context: Overview, Problem Statement and Goals", draft-ietf-6tisch-tsch-00 (work in progress), November 2013.

[I-D.wang-6tisch-6top]

Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top)", draft-wang-6tisch-6top-00 (work in progress), October 2013.

## 10.2. 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.

## Authors' Addresses

Diego Dujovne (editor)  
Universidad Diego Portales  
Escuela de Informatica y Telecomunicaciones  
Av. Ejercito 441  
Santiago, Region Metropolitana  
Chile

Phone: +56 (2) 676-8121  
Email: diego.dujovne@mail.udp.cl

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

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

Nicola Accettura  
University of California Berkeley  
Berkeley Sensor & Actuator Center  
490 Cory Hall  
Berkeley, California 94720  
USA

Email: nicola.accettura@eecs.berkeley.edu

6TiSCH  
Internet-Draft  
Intended status: Informational  
Expires: January 5, 2015

Q. Wang, Ed.  
Univ. of Sci. and Tech. Beijing  
X. Vilajosana  
Universitat Oberta de Catalunya  
T. Watteyne  
Linear Technology  
July 4, 2014

6TiSCH Operation Sublayer (6top) Interface  
draft-ietf-6tisch-6top-interface-01

Abstract

This document defines a generic data model for the 6TiSCH Operation Sublayer (6top), using the YANG data modeling language. This data model can be used for future network management solutions defined by the 6TiSCH working group. This document also defines a list commands internal to the 6top sublayer.

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 5, 2015.

## Copyright Notice

Copyright (c) 2014 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. 6TiSCH Operation Sublayer (6top) Overview . . . . .	3
2.1. Cell Model . . . . .	4
2.1.1. hard cells . . . . .	6
2.1.2. soft cells . . . . .	6
2.2. Data Transfer Model . . . . .	6
3. Generic Data Model . . . . .	8
3.1. YANG model of the 6top MIB . . . . .	8
3.2. YANG model of the IEEE802.15.4 PIB . . . . .	24
4. Commands . . . . .	29
5. References . . . . .	32
5.1. Normative References . . . . .	33
5.2. Informative References . . . . .	33
5.3. External Informative References . . . . .	34
Authors' Addresses . . . . .	34

## 1. Introduction

This document defines a generic data model for the 6TiSCH Operation Sublayer (6top), using the YANG data modeling language defined in [RFC6020]. This data model can be used for future network management solutions defined by the 6TiSCH working group. This document also defines a list commands internal to the 6top sublayer. This data model gives access to metrics (e.g. cell state), TSCH configuration and control procedures, and support for the different scheduling mechanisms described in [I-D.ietf-6tisch-architecture]. The 6top sublayer addresses the set of management information and functionalities described in [I-D.ietf-6tisch-tsch].

For example, network formation in a TSCH network is handled by the use of Enhanced Beacons (EB). EBs include information for joining

nodes to be able to synchronize and set up an initial network topology. However, [IEEE802154e] does not specify how the period of EBs is configured, nor the rules for a node to select a particular node to join. 6top offers a set of commands so control mechanisms can be introduced on top of TSCH to configure nodes to join a specific node and obtain a unique 16-bit identifier from the network. Once a network is formed, 6top maintains the network's health, allowing for nodes to stay synchronized. It supplies mechanisms to manage each node's time source neighbor and configure the EB interval. Network layers running on top of 6top take advantage of the TSCH MAC layer information so routing metrics, topological information, energy consumption and latency requirements can be adjusted to TSCH, and adapted to application requirements.

TSCH requires a mechanism to manage its schedule; 6top provides a set of commands for upper layers to set up specific schedules, either explicitly by detailing specific cell information, or by allowing 6top to establish a schedule given a bandwidth or latency requirement. 6top is designed to enable decentralized, centralized or hybrid scheduling solutions. 6top enables internal TSCH queuing configuration, size of buffers, packet priorities, transmission failure behavior, and defines mechanisms to encrypt and authenticate MAC slotframes.

As described in [morell04label], due to the slotted nature of a TSCH network, it is possible to use a label switched architecture on top of TSCH cells. As a cell belongs to a specific track, a label header is not needed at each packet; the input cell (or bundle) and the output cell (or bundle) uniquely identify the data flow. The 6top sublayer provides operations to manage the cell mappings.

## 2. 6TiSCH Operation Sublayer (6top) Overview

6top is a sublayer which is the next-higher layer for TSCH (Figure 1), as detailed in [I-D.ietf-6tisch-architecture]. 6top offers both management and data interfaces to an upper layer, and includes monitoring and statistics collection, both of which are configurable through its management interface. The detail of 6top-sublayer is described in [I-D.wang-6tisch-6top-sublayer]



## Protocol Stack

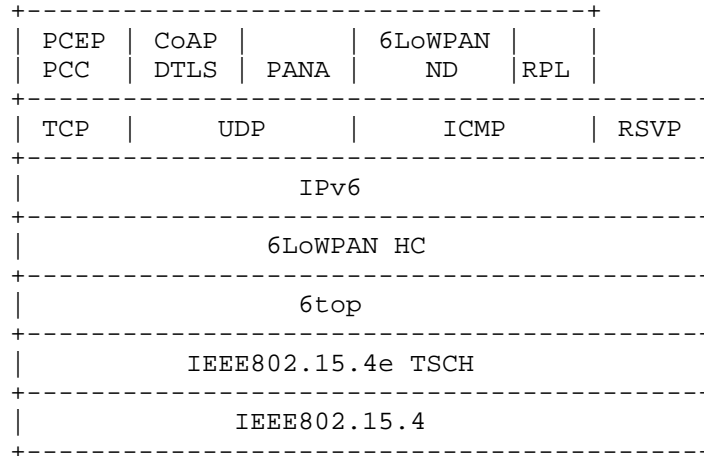


Figure 1

6top distinguishes between hard cells and soft cells. It therefore requires an extra flag to all cells in the TSCH schedule, as detailed in Section 2.1.

When a higher layer gives 6top a 6LoWPAN packet for transmission, 6top maps it to the appropriate outgoing priority-based queue, as detailed in Section 2.2.

Section 3 contains a generic data model for the 6top sublayer, described in the YANG data modeling language.

The commands of the management and data interfaces are listed in Section 4. This set of commands is designed to support decentralized, centralized and hybrid scheduling solutions.

## 2.1. Cell Model

[IEEE802154e] defines a set of options attached to each cell. A cell can be a Transmit cell, a Receive cell, a Shared cell or a Timekeeping cell. These options are not exclusive, as a cell can be qualified with more than one of them. The MLME-SET-LINK.request command defined in [IEEE802154e] uses a linkOptions bitmap to specify the options of a cell. Acceptable values are:

b0 = Transmit

b1 = Receive

b2 = Shared

b3 = Timekeeping

b4-b7 = Reserved

Only Transmit cells can also be marked as Shared cells. When the shared bit is set, a back-off procedure is applied to handle collisions. Shared behavior does not apply to Receive cells.

6top allows an upper layer to schedule a cell at a specific slotOffset and channelOffset, in a specific slotframe.

In addition, 6top allows an upper layer to schedule a certain amount of bandwidth to a neighbor, without having to specify the exact slotOffset(s) and channelOffset(s). Once bandwidth is reserved, 6top is in charge of ensuring that this requirement is continuously satisfied. 6top dynamically reallocates cells if needed, and over-provisions if required.

6top allows an upper layer to associate a cell with a specific track by using a TrackID. A TrackID is a tuple (TrackOwnerAddr, InstanceID), where TrackOwnerAddr is the address of the node which initializes the process of creating the track, i.e., the owner of the track; and InstanceID is an instance identifier given by the owner of the track. InstanceID comes from upper layer; InstanceID could for example be the local instance ID defined in RPL.

If the TrackID is set to (0,0), the cell can be used by the best-effort QoS configuration or as a Shared cell. If the TrackID is not set to (0,0), i.e., the cell belongs to a specific track, the cell MUST not be set as Shared cell.

6top allows an upper layer to ask a node to manage a portion of a slotframe, which is named as chunk. Chunks can be delegated explicitly by the PCE to a node, or claimed automatically by any node that participates to the distributed cell scheduling process. The resource in a chunk can be appropriated by the node, i.e. the owner of the chunk.

Given this mechanism, 6top defines hard cells (which have been requested specifically) and soft cells (which can be reallocated dynamically). The hard/soft flag is introduced by the 6top sublayer named as CellType, 0: soft cell, 1: hard cell. This option is mandatory; all cells are either hard or soft.

#### 2.1.1. hard cells

A hard cell is a cell that cannot be dynamically reallocated by 6top. The CellType MUST be set to 1. The cell is installed by 6top given specific slotframe ID, slotOffset, and channelOffset.

#### 2.1.2. soft cells

A soft cell is a cell that can be reallocated by 6top dynamically. The CellType MUST be set to 0. This cell is installed by 6top given a specific bandwidth requirement. Soft cells are installed through the soft cell negotiation procedure described in [I-D.wang-6tisch-6top-sublayer].

### 2.2. Data Transfer Model

Once a TSCH schedule is established, 6top is responsible for feeding the data from the upper layer into TSCH. This section describes how 6top shapes data from the upper layer (e.g., RPL, 6LoWPAN), and feeds it to TSCH. Since 6top is a sublayer between TSCH and 6LoWPAN, the properties associated with a packet/fragment from the upper layer includes the next hop neighbor (DestAddr) and expected sending priority of the packet (Priority), and/or TrackID(s). The output to TSCH is the fragment corresponding to the next active cell in the TSCH schedule.

## 6top Data Transfer Model

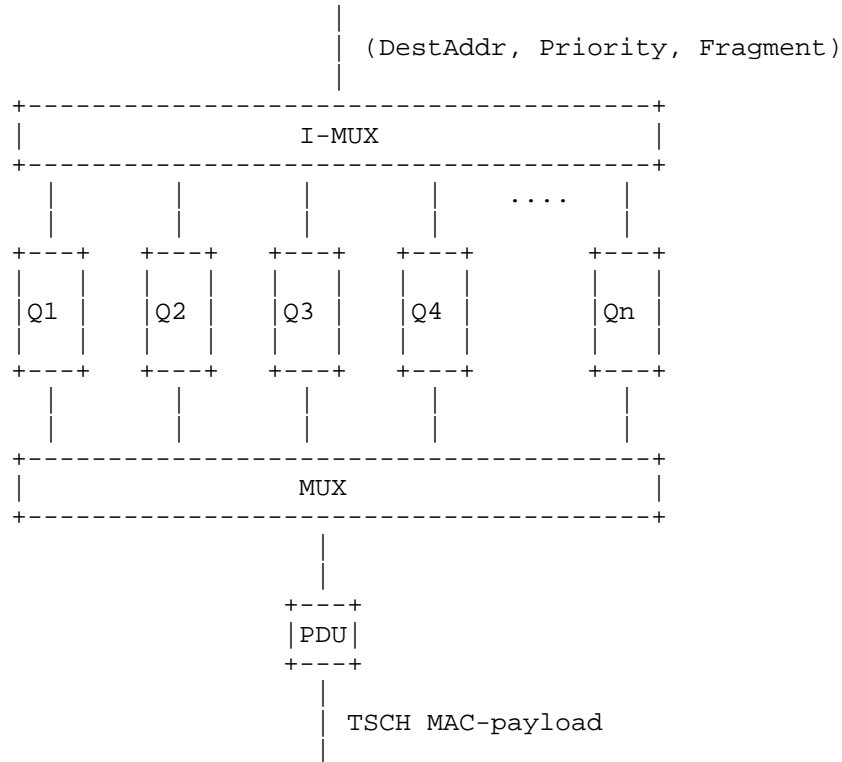


Figure 2

In Figure 2,  $Q_i$  represents a queue, which is either broadcast or unicast, and has an assigned priority. The number of queues is configurable. The relationship between queues and tracks is configurable. For example, for a given queue, only one specific track can be used, all of the tracks can be used, or a subset of the tracks can be used.

When 6top receives a packet to transmit through a `Send.data` command (Section 4), the `I-MUX` module selects a queue in which to insert it. If the packet's destination address is a unicast (resp. broadcast) address, it will be inserted into a unicast (resp. broadcast) queue.

The `MUX` module is invoked at each scheduled transmit cell by TSCH. When invoked, the `MUX` module goes through the queues, looking for the best matching frame to send. If it finds a frame, it hands it over to TSCH for transmission. If the next active cell is a broadcast cell, it selects a fragment only from broadcast queues.

How the MUX module selects the best frame is configurable. The following rules are a typical example:

The frame's layer 2 destination address MUST match the neighbor address associated with the transmit cell.

If the transmit cell is associated with a specific track, the frames in the queue corresponding to the TrackID have the highest priority.

If the transmit cell is not associated with a specific track, i.e., TrackID=(0,0), frames from a queue with a higher priority MUST be sent before frames from a queue with a lower priority.

Further rules can be configured to satisfy specific QoS requirements.

### 3. Generic Data Model

This section presents the generic data model of the 6top sublayer, using the YANG data modeling language. This data model can be used for future network management solutions defined by the 6TiSCH working group. The data model consists of the MIB (management information base) defined in 6top, and part of the PIB (personal area network information base) defined in [IEEE802154e] and [IEEE802154].

#### 3.1. YANG model of the 6top MIB

```
list CellList {
  key "CellID";
  description
    "List of scheduled cells of a node with all of its neighbors,
    in all of its slotframes.";

  leaf CellID {
    type uint16;
    description
      "Equal to Linkhandle in the linkTable of TSCH";
    reference
      "IEEE802154e";
  }
  leaf SlotframeID {
    type uint8;
    description
      "SlotframeID, one in SlotframeList, indicates the slotframe
      the cell belongs to.";
    reference
      "IEEE802154e";
  }
}
```

```
leaf SlotOffset {
    type uint16;
    description
        "Defined in IEEE802154e.";
    reference
        "IEEE802154e";
}
leaf ChannelOffset {
    type uint8;
    description
        "Defined in IEEE802154e. Value range is 0..15";
    reference
        "IEEE802154e";
}
leaf LinkOption {
    type bits {
        bit Transmit {
            position 0;
        }
        bit Receive {
            position 1;
        }
        bit Share {
            position 2;
        }
        bit Timekeeping {
            position 3;
        }
        bit Reserved1 {
            position 4;
        }
        bit Reserved2 {
            position 5;
        }
        bit Reserved3 {
            position 6;
        }
        bit Reserved4 {
            position 7;
        }
    }
    description
        "Defined in IEEE802154e.";
    reference
        "IEEE802154e";
}
leaf LinkType {
    type enumeration {
```

```
        enum NORMAL;
        enum ADVERTISING;
    }
    description
    "Defined in IEEE802154";
    reference
    "IEEE802154";
}
leaf CellType {
    type enumeration {
        enum SOFT;
        enum HARD;
    }
    description
    "Defined in 6top";
}
leaf TargetNodeAddress {
    type uint64;
    description
    "Defined by 6top, but being constrained by TSCH
    macNodeAddress size, 2-octets. If using TSCH as MAC,
    higher 6-octets should be filled with 0, and lowest
    2-octets is neighbor address";
}
leaf TrackID {
    type uint16;
    description
    "A TrackID is one in the TrackList, pointing to a tuple
    (TrackOwnerAddr,InstanceID) , where TrackOwnerAddr is the
    address of the node which initializes the process of
    creating the track, i.e., the owner of the track; and
    InstanceID is an instance identifier given by the owner of
    the track.";
}
container Statistic {
    leaf NumOfStatistic {
        type uint8;
        description
        "Number of statistics collected on the cell";
    }
    list MeasureList {
        key "StatisticsMetricsID";
        leaf StatisticsMetricsID{
            type uint16;
            description
            "An index of StatisticsMetricList, which defines how
            to collect data and get the statistice value";
        }
    }
}
```

```
        leaf StatisticsValue{
            type uint16;
            config false;
            description
                "updated by 6top according to the statistics method
                specified by StatisticsMetricsID";
        }
    }
}

list SlotframeList {
    key "SlotframeID";
    description
        "List of all of the slotframes used by the node.";

    leaf SlotframeID {
        type uint8;
        description
            "Equal to SlotframeHandle defined in TSCH";
        reference
            "IEEE802154e";
    }
    leaf NumOfSlots {
        type uint16;
        description
            "indicates how many timeslots in the slotframe";
    }
}

list MonitoringStatusList {
    key "MonitoringStatusID";
    description
        "List of the monitoring configuration and results per
        slotframe and neighbor. Basically, it is used for Monitoring
        Function of 6top to re-allocate softcells or initial the
        softcell negotiation process to increase/decrease number of
        softcells. Upper layer can use it also.";

    leaf MonitoringStatusID {
        type uint16;
    }
    leaf SlotframeID {
        type uint8;
        description
            "SlotframeID, one in SlotframeList, indicates the slotframe
            being monitored";
        reference

```



```
    "IEEE802154e";
  }
  leaf TargetNodeAddress {
    type uint64;
    description
      "Defined by 6top, but being constrained by TSCH
      macNodeAddress size, 2-octets. If using TSCH as MAC,
      higher 6-octets should be filled with 0, and lowest
      2-octets is neighbor address. It indicates the communication
      link being monitored";
  }
  leaf EnforcePolicy {
    type enumeration {
      enum DISABLE;
      enum BESTEFFORT;
      enum STRICT;
      enum OVERPROVISION;
    }
    description
      "Currently enforced QoS policy. DISABLE-no QoS;
      BESTEFFORT- best effort policy is used; STRICT- Strict
      Priority Queueing; OVERPROVISION- cell overprovision";
  }
  leaf AllocatedHard {
    type uint16;
    config false;
    description
      "Number of hard cells allocated";
  }
  leaf AllocatedSoft {
    type uint16;
    config false;
    description
      "Number of soft cells allocated";
  }
  leaf OverProvision {
    type uint16;
    config false;
    description
      "Overprovisioned cells. 0 if EnforcePolicy is
      DISABLE";
  }
  leaf QoS {
    type uint16;
    config false;
    description
      "Current QoS including overprovisioned cells, i.e. the
      bandwidth obtained including the overprovisioned cells.";
  }
}
```

```
    }
    leaf NQoS {
        type uint16;
        config false;
        description
            "Real QoS without over provisioned cells, i.e. the actual
            bandwidth without taking into account the overprovisioned
            cells.";
    }
}

list StatisticsMetricsList {
    key "StatisticsMetricsID";
    description
        "List of Statistics Metrics used in the node.";

    leaf StatisticsMetricsID {
        type uint16;
    }
    leaf SlotframeID {
        type uint16;
        description
            "SlotframeID, one in SlotframeList, specifies the slotframe to
            which the statistics metrics applies to. If empty, applies to
            all slotframes";
        reference
            "IEEE802154e";
    }
    leaf SlotOffset {
        type uint16;
        description
            "Specific slotOffset to which the statistics metrics applies
            to. If empty, applies to all timeslots";
        reference
            "IEEE802154e";
    }
    leaf ChannelOffset {
        type uint8;
        description
            "Specific channelOffset to which the statistics metrics applies
            to. If empty, applies to all channels";
        reference
            "IEEE802154e";
    }
    leaf TargetNodeAddress {
        type uint64;
        description
            "Specific neighbor nodes to which the statistics metrics
```

```
        applies to. If empty, applies to all neighbor nodes.";
    }
    leaf Metrics {
        type enumeration {
            enum PDR;
            enum ETX;
            enum RSSI;
            enum LQI;
        }
        description
            "The metric to be monitored.";
    }
    leaf Window {
        type uint16;
        description
            "measurement period, in Number of the slotframe size";
    }
    leaf Enable {
        type enumeration {
            enum DISABLE;
            enum ENABLE;
        }
        description
            "indicates the StatisticsMetric is active or not";
    }
}
```

```
list EBList {
  key "EbID";
  description
    "List of information related with the EBs used by the node";

  leaf EbID {
    type uint8;
  }
  leaf CellID {
    type uint16;
    description
      "CellID, one in CellList, indicates the cell used to send
      EB";
  }
  leaf Peroid {
    type uint16;
    description
      "The EBs period, in seconds, indicates the interval between
      two EB sendings";
  }
  leaf Expiration {
    type enumeration {
      enum NEVERSTOP;
      enum EXPIRATION;
    }
    description
      "NEVERSTOP- the period of the EB never stops; EXPIRATION-
      when the Period arrives, the EB will stop.";
  }
  leaf Priority {
    type uint8;
    description
      "The joining priority model that will be used for
      advertisements. Joining priority MAY be for example
      SAME_AS_PARENT, RANDOM, BEST_PARENT+1 or
      DAGRANK(rank).";
  }
}
```

```
container TimeSource {
  description
    "specify the timesource selection policy and some relative
    statistics.";

  leaf policy {
    type enumeration {
      enum ALLPARENT;
      enum BESTCONNECTED;
      enum LOWESTJOINPRIORITY;
    }
    description
      "indicates the policy to choose timesource. ALLPARENT- choose
      from all parents; BESTCONNECTED- choose the best-connected
      node; LOWESTJOINPRIORITY- choose the node with lowest priority
      in its EB.";
  }
  leaf TargetNodeAddress {
    type uint64;
    description
      "Address of the time source neighbor";
  }
  leaf MinTimeCorrection {
    type uint16;
    config false;
    description
      "measured in microsecond";
  }
  leaf MaxTimeCorrection {
    type uint16;
    config false;
    description
      "measured in microsecond";
  }
  leaf AveTimeCorrection {
    type uint16;
    config false;
    description
      "measured and computed in microsecond";
  }
}
```

```
typedef asntype {
    description
        "The type to store ASN. String of 5 bytes";
    type string {
        length "0..5";
    }
}

list NeighborList {
    key "TargetNodeAddress";
    description
        "statistics per communication link.";

    leaf TargetNodeAddress {
        type uint64;
        description
            "Address of the time source neighbor";
    }
    leaf RSSI {
        type uint8;
        config false;
        description
            "The received signal strength";
    }
    leaf LinkQuality {
        type uint8;
        config false;
        description
            "The LQI metric";
    }
    leaf ASN {
        type asntype;
        config false;
        description
            "The 5 ASN bytes, indicates the most recent timeslot when a
            packet from the neighbor was received";
    }
}

list QueueList {
    key "QueueId";
    description
        "List of Queues, including configuration and statistics.";

    leaf QueueId {
        type uint8;
        description
            "Queue Identifier";
    }
}
```

```
}
leaf TxqLength {
    type uint8;
    description
        "The TX queue length in number of packets";
}
leaf RxqLength {
    type uint8;
    description
        "The RX queue length in number of packets";
}
leaf NumrTx {
    type uint8;
    description
        "Number of allowed retransmissions.";
}
leaf Age {
    type uint16;
    description
        "In seconds. Discard packet according to its age
        on the queue. 0 if no discards are allowed.";
}
leaf RTXbackoff {
    type uint8;
    description
        "retransmission backoff in number of slotframes.
        0 if next available timeslot wants to be used.";
}
leaf StatsWindow {
    type uint16;
    description
        "In second, window of time used to compute stats.";
}
leaf QueuePriority {
    type uint8;
    description
        "The priority for this queue.";
}
list TrackIds {
    key "TrackID";
    leaf TrackID{
        type uint16;
        description
            "The TrackID, one in TrackList, indicates the Track is
            associated with the Queue.";
    }
}
leaf MinLenTXQueue {
```

```
        type uint8;
        config false;
        description
            "Statistics, lowest TX queue length registered in the window.";
    }
    leaf MaxLenTXQueue {
        type uint8;
        config false;
        description
            "Statistics, largest TX queue length registered in the
            window.";
    }
    leaf AvgLenTXQueue {
        type uint8;
        config false;
        description
            "Statistics, avg TX queue length registered in the window.";
    }
    leaf MinLenRXQueue {
        type uint8;
        config false;
        description
            "Statistics, lowest RX queue length registered in the window.";
    }
    leaf MaxLenRXQueue {
        type uint8;
        config false;
        description
            "Statistics, largest RX queue len registered in the window.";
    }
    leaf AvgLenRXQueue {
        type uint8;
        config false;
        description
            "Statistics, avg RX queue length registered in the window.";
    }
    leaf MinRetransmissions {
        type uint8;
        config false;
        description
            "Statistics, lowest number of retransmissions registered in
            the window.";
    }
    leaf MaxRetransmissions {
        type uint8;
        config false;
        description
            "Statistics, largest number of retransmissions registered
```



```
        in the window.";
    }
    leaf AvgRetransmissions {
        type uint8;
        config false;
        description
            "Statistics, average number of retransmissions registered
            in the window.";
    }
    leaf MinPacketAge {
        type uint16;
        config false;
        description
            "Statistics, in seconds, minimum time a packet stayed in
            the queue during the observed window.";
    }
    leaf MaxPacketAge {
        type uint16;
        config false;
        description
            "Statistics, in seconds, maximum time a packet stayed
            in the queue during the observed window.";
    }
    leaf AvgPacketAge {
        type uint16;
        config false;
        description
            "Statistics, in seconds, average time a packet stayed in
            the queue during the observed window.";
    }
    leaf MinBackoff {
        type uint8;
        config false;
        description
            "Statistics, in number of slotframes, minimum Backoff
            for a packet in the queue during the observed window.";
    }
    leaf MaxBackoff {
        type uint8;
        config false;
        description
            "Statistics, in number of slotframes, maximum Backoff
            for a packet in the queue during the observed window.";
    }
    leaf AvgBackoff {
        type uint8;
        config false;
        description
```

```
        "Statistics, in number of slotframes, average Backoff
        for a packet in the queue during the observed window.";
    }
}

list LabelSwitchList {
    key "LabelSwitchID";
    description
    "List of Label switch' configuration on the node";

    leaf LabelSwitchID {
        type uint16;
    }
    list InputCellIds {
        key "CellID";
        leaf CellID{
            type uint16;
            description
            "The CellID, indicates the Rx cell on which the packet will
            come in.";
        }
    }
    list OutputCellIds {
        key "CellID";
        leaf CellID{
            type uint16;
            description
            "The CellID, indicates the Tx cell on which the received
            packet should be sent out.";
        }
    }
    leaf LoadBalancingPolicy {
        type enumeration {
            enum ROUNDROBIN;
            enum OTHER;
        }
        description
        "The load-balancing policy. ROUNDROBIN- Round robin algorithm
        is used for forwarding scheduling.";
    }
}
```

```
list TrackList {
  key "TrackId";
  description
    "List of the tracks through the node.";

  leaf TrackId {
    type uint16;
    description
      "Track Identifier, named locally. It is used to refer to the
      tuple (TrackOwnerAddr, InstanceID).";
  }
  leaf TrackOwnerAddr {
    type uint64;
    description
      "The address of the node which initializes the process of
      creating the track, i.e., the owner of the track;";
  }
  leaf InstanceID {
    type uint16;
    description
      "InstanceID is an instance identifier given by the owner of
      the track. InstanceID comes from upper layer; InstanceID could
      for example be the local instance ID defined in RPL.";
  }
}
```

```
list ChunkList {
  key "ChunkId";
  description
    "List of the chunks assigned to the node.";

  leaf ChunkId{
    type uint16;
    description
      "The identifier of a chunk";
  }
  leaf SlotframeId{
    type uint8;
    description
      "SlotframeID, one in SlotframeList, indicates the
      slotframe to which the chunk belongs";
  }
  leaf SlotBase {
    type uint16;
    description
      "the base slotOffset of the chunk in the slotframe";
  }
  leaf SlotStep {
    type uint8;
    description
      "the slot incremental of the chunk";
  }
  leaf ChannelBase {
    type uint8;
    description
      "the base channelOffset of the chunk";
  }
  leaf ChannelStep {
    type uint8;
    description
      "the channel incremental of the chunk";
  }
  leaf ChunkSize {
    type uint8;
    description
      "the number of cells in the chunk. The chunk is the set
      of (slotOffset(i), channelOffset(i)),
      i=0..Chunksize-1,
      slotOffset(i)= (slotBase + i * slotStep) % slotframeLen,
      channelOffset(i) = (channelBase + i * channelStep) % 16";
  }
}
```

```

list ChunkCellList {
  key "SlotOffset ChannelOffset";
  description
    "List of all of the cells assigned to the node via the
    assignment of chunks.";

  leaf SlotOffset{
    type uint16;
    description
      "The slotoffset of a cell which belongs to a Chunk";
  }
  leaf ChannelOffset{
    type uint16;
    description
      "The channeloffset of a cell which belongs to a chunk.";
  }
  leaf ChunkId {
    type uint16;
    description
      "Identifier of the chunk the cell belongs to";
  }
  leaf CellID{
    type uint16;
    description
      "Initial value of CellID is 0xFFFF. When the cell is
      scheduled, the value of CellID is same as that in
      CellList";
  }
  leaf ChunkCellStatus {
    type enumeration {
      enum UNSCHEDULED;
      enum SCHEDULED;
    }
  }
}

```

### 3.2. YANG model of the IEEE802.15.4 PIB

This section describes the YANG model of the part of PIB ([IEEE802154] and [IEEE802154e]) used by 6top, such as security related attributes, TSCH related attributes. This part of data will be accessed through the MLME-GET and MLME-SET primitive [IEEE802154] directly, instead of using 6top commands.

TODO the security related attributes will be added after 6TisCH WG has consensus on the security scheme of 6top

```

container TSCHSpecificPIBAttributes {

```

```
description
"TSCH specific MAC PIB attributes.";
reference
"table 52b in IEEE802.15.4e-2012.";

leaf macMinBE {
    type uint8;
    description
    "defined in Table 52b of IEEE802.15.4e-2012,
    The minimum value of the backoff exponent (BE) in the
    CSMA-CA algorithm or the TSCH-CA algorithm. default:
    3-CSMA-CA, 1-TSCH-CA";
}
leaf macMaxBE {
    type uint8;
    description
    "defined in Table 52b of IEEE802.15.4e-2012,
    The maximum value of the backoff exponent (BE) in the
    CSMA-CA algorithm or the TSCH-CA algorithm. default:
    5-CSMA-CA, 7-TSCH-CA";
}
leaf macDisconnectTime {
    type uint16;
    description
    "defined in Table 52b of IEEE802.15.4e-2012,
    Time (in Timeslots) to send out Disassociate frames
    before disconnecting, default: 0x00ff";
}
leaf macJoinPriority {
    type uint8;
    description
    "defined in Table 52b of IEEE802.15.4e-2012,
    The lowest join priority from the TSCH Synchronization
    IE in an Enhanced beacon, default: 1";
}
leaf macASN {
    type asntype;
    description
    "defined in Table 52b of IEEE802.15.4e-2012,
    The Absolute Slot Number, i.e., the number of slots
    that ha elapsed since the start of the network.";
}
leaf macNoHLBuffers {
    type enumeration {
        enum TRUE;
        enum FALSE;
    }
    description
```

```
        "defined in Table 52b of IEEE802.15.4e-2012,  
        If the value is TRUE, the higher layer receiving the  
        frame payload cannot buffer it, and the device should  
        acknowledge frames with a NACK; If FALSE, the higher  
        layer can accept the frame payload. default: FALSE";  
    }  
}  
  
list TSCHmacTimeslotTemplate {  
    key "macTimeslotTemplateId";  
    description  
        "List of all timeslot templates used in the node.";  
    reference  
        "table 52e in IEEE802.15.4e-2012.";  
  
    leaf macTimeslotTemplateId {  
        type uint8;  
        description  
            "defined in Table 52e of IEEE802.15.4e-2012.  
            Identifier of Timeslot Template. default: 0";  
    }  
    leaf macTsCCAOffset {  
        type uint16;  
        description  
            "The time between the beginning of timeslot and start  
            of CCA operation, in microsecond. default: 1800";  
    }  
    leaf macTsCCA {  
        type uint16;  
        description  
            "Duration of CCA, in microsecond. default: 128";  
    }  
    leaf macTsTxOffset {  
        type uint16;  
        description  
            "The time between the beginning of the timeslot and  
            the start of frame transmission, in microsecond.  
            default: 2120";  
    }  
    leaf macTsRxOffset {  
        type uint16;  
        description  
            "Beginning of the timeslot to when the receiver shall  
            be listening, in microsecond. default: 1120";  
    }  
    leaf macTsRxAckDelay {  
        type uint16;  
        description
```

```
        "End of frame to when the transmitter shall listen for
        Acknowledgment, in microsecond. default: 800";
    }
    leaf macTsTxAckDelay {
        type uint16;
        description
            "End of frame to start of Acknowledgment, in
            microsecond.
            default: 1000";
    }
    leaf macTsRxWait {
        type uint16;
        description
            "The time to wait for start of frame, in microsecond.
            default: 2200";
    }
    leaf macTsAckWait {
        type uint16;
        description
            "The minimum time to wait for start of an
            Acknowledgment, in microsecond. default: 400";
    }
    leaf macTsRxTx {
        type uint16;
        description
            "Transmit to Receive turnaround, in microsecond.
            default: 192";
    }
    leaf macTsMaxAck {
        type uint16;
        description
            "Transmission time to send Acknowledgment, in
            microsecond. default: 2400";
    }
    leaf macTsMaxTx {
        type uint16;
        description
            "Transmission time to send the maximum length frame,
            in microsecond. default: 4256";
    }
    leaf macTsTimeslotLength {
        type uint16;
        description
            "The total length of the timeslot including any unused
            time after frame transmission and Acknowledgment,
            in microsecond. default: 10000";
    }
}
```



```
list TSCHHoppingSequence {
  key "macHoppingSequenceID";
  description
    "List of all channel hopping sequences used in the
    nodes";
  reference
    "Table 52f of IEEE802.15.4e-2012";

  leaf macHoppingSequenceID {
    type uint8;
    description
      "defined in Table 52f of IEEE802.15.4e-2012.
      Each hopping sequence has a unique ID. default: 0";
  }
  leaf macChannelPage {
    type uint8;
    description
      "Corresponds to the 5 MSBs (b27, ..., b31) of a row
      in phyChannelsSupported. Note this may not correspond
      to the current channelPage in use.";
  }
  leaf macNumberOfChannels {
    type uint16;
    description
      "Number of channels supported by the PHY on this
      channelPage.";
  }
  leaf macPhyConfiguration {
    type uint32;
    description
      "For channel pages 0 to 6, the 27 LSBs(b0, b1, ...,
      b26) indicate the status (1 = to be used, 0 = not to
      be used) for each of the up to 27 valid channels
      available to the PHY. For pages 7 and 8, the 27 LSBs
      indicate the configuration of the PHY, and the channel
      list is contained in the extendedBitmap.";
  }
  leaf macExtendedBitmap {
    type uint64;
    description
      "For pages 7 and 8, a bitmap of numberOfChannels bits,
      where bk shall indicate the status of channel k for
      each of the up to numberOfChannels valid channels
      supported by that channel page and phyConfiguration.
      Otherwise field is empty.";
  }
  leaf macHoppingSequenceLength {
    type uint16;
  }
}
```

```
        description
        "The number of channels in the Hopping Sequence.
        Does not necessarily equal numberOfChannels.";
    }
    list macHoppingSequenceList {
        key "HoppingChannelID";
        leaf HoppingChannelID {
            type uint16;
            description
            "channels to be hopped over";
        }
    }
    leaf macCurrentHop {
        type uint16;
        config false;
        description
        "Index of the current position in the hopping sequence
        list.";
    }
}
```

#### 4. Commands

6top provides a set of commands as the interface with the higher layer. Most of these commands are related to the management of slotframes, cells and scheduling information. 6top also provides an interface allowing an upper layer to retrieve status information and statistics. The command set aims to facilitate 6top implementation by describing the main operations that higher layers may use to interact with 6top. The listed commands aim at providing semantics to manipulate 6top MIB, IEEE802.15.4 PIB and IEEE802.15.4e PIB programmatically.

**CREATE.hardcell:** Creates one or more hard cells in the schedule. Fails if the cell already exists. A cell is uniquely identified by the tuple (slotframe ID, slotOffset, channelOffset). 6top schedules the cell and marks it as a hard cell, indicating that it cannot reschedule this cell. The return value is CellID and the created cell is also filled in CellList(Section 3.1).

**CREATE.softcell:** To create soft cell(s). 6top is responsible for picking the exact slotOffset and channelOffset in the schedule, and ensure that the target node chooses the same cell and TrackID. 6top marks these cells as soft cell, indicating that it will continuously monitor their performance and reschedule if needed. The return value is CellID, and the created cell is also filled in CellList (Section 3.1).

READ.cell: Given a (slotframe ID, slotOffset, channelOffset), retrieves the cell information. A read command can be issued for any cell, hard or soft. 6top gets cell information from CellList (Section 3.1).

UPDATE.cell: Update a hard cell, i.e., re-allocate it to a different slotOffset and/or channelOffset. Fails if the cell does not exist. CellList (Section 3.1) will be modified.

DELETE.hardcell: To remove a hard cell. This removes the hard cell from the node's schedule, from CellList (Section 3.1).

DELETE.softcell: To remove a (number of) soft cell(s). This command leads the pair of nodes figure out the specific cell(s) to be removed. After that, the cell(s) will be removed from the CellLists (Section 3.1) on both sides.

REALLOCATE.softcell: To force a re-allocation of a soft cell. The reallocated cell will be installed in a different slotOffset, channelOffset but slotframe and TrackID remain the same. Hard cells MUST NOT be reallocated. This command will result in the modification of CellLists (Section 3.1) on both sides.

CREATE.slotframe: Creates a new slotframe. Adds a entry to the SlotframeList (Section 3.1).

READ.slotframe: Returns the information of a slotframe given its slotframeID from SlotframeList (Section 3.1).

UPDATE.slotframe: Change the number of timeslots in a slotframe given its slotframeID in SlotframeList (Section 3.1).

DELETE.slotframe: Deletes a slotframe, remove it from SlotframeList (Section 3.1).

CONFIGURE.monitoring: Configures the level of QoS the Monitoring process MUST enforce, i.e. config MonitoringStatusList (Section 3.1).

READ.monitoring: Reads the current Monitoring status from MonitoringStatusList (Section 3.1).

CONFIGURE.statistics: Configures the statistics process in StatisticsMetricsList (Section 3.1). The CONFIGURE.statistics enables flexible configuration and supports empty parameters that will force 6top to conduct statistics on all members of that dimension. For example, if ChannelOffset is empty and metric is

set as PDR, then, 6top will conduct the statistics of PDR on all of channels.

READ.statistics: Reads a metric for the specified dimension. Information is aggregated according to the parameters from CellList (Section 3.1).

RESET.statistics: Resets the gathered statistics in CellList (Section 3.1).

CONFIGURE.eb: Configures EBs, i.e. configures EBlist (Section 3.1).

READ.eb: Reads the EBs configuration from EBList (Section 3.1).

CONFIGURE.timesource: Configures the Time Source Neighbor selection process, i.e. configure TimeSource (Section 3.1).

READ.timesource: Retrieves information about the time source neighbors of that node from TimeSource (Section 3.1).

CREATE.neighbor: Creates an entry for a neighbor in the neighbor table, i.e. NeighborList (Section 3.1).

READ.all.neighbor: Returns the list of neighbors of that node according to NeighborList (Section 3.1).

READ.neighbor: Returns the information of a specific neighbor of that node specified by its neighbor address according to NeighborList (Section 3.1).

UPDATE.neighbor: Updates the last status for a given TargetNodeAddress in the NeighborList (Section 3.1).

DELETE.neighbor: Deletes a neighbor given its address from NeighborList (Section 3.1).

CREATE.queue: Creates and Configures a queue in QueueList (Section 3.1).

READ.queue: Reads the queue configuration for given QueueId from QueueList (Section 3.1).

READ.queue.stats: For a given QueueId, reads the queue statistics information from the QueueList (Section 3.1).

UPDATE.queue: For a given QueueId, update its configuration in the QueueList (Section 3.1).

DELETE.queue: Deletes a Queue for a given QueueId from the QueueList (Section 3.1).

LabelSwitching.map: Maps an input cell or a bundle of input cells to an output cell or a bundle of output cells, i.e. adds a entry to the LabelSwitchList (Section 3.1).

LabelSwitching.unmap: Unmap one input cell or a bundle of input cells to an output cell or a bundle of output cells, i.e. modifies the LabelSwitchList (Section 3.1).

CREATE.chunk: Creates a chunk which consists of one or more unscheduled cells, i.e. add an entry to the ChunkList (Section 3.1).

READ.chunk: Returns the information of a chunk given its ChunkID from ChunkList (Section 3.1).

DELETE.chunk: For given ChunkId, removes a chunk from the ChunkList (Section 3.1), which also causes all of the scheduled cells in the chunk to be deleted from the TSCH schedule and CellList (Section 3.1).

CREATE.hardcell.fromchunk: Creates one or more hard cells from a chunk. 6top schedules the cell and marks it as a hard cell, indicating that it cannot reschedule this cell. The cell will be added into the CellList (Section 3.1). In addition, 6top will change the attributes corresponding to the cell in the ChunkCellList (Section 3.1), i.e. its CellID is changed to the same CellID in the CellList, and its Status is changed to SCHEDULED.

READ.chunkcell: Returns the information of all cells in a chunk given its ChunkID from ChunkCellList (Section 3.1).

DELETE.hardcell.fromchunk: To remove a hard cell which comes from a chunk. This removes the hard cell from the node's schedule and CellList (Section 3.1). In addition, it changes the attributes corresponding to the cell in the ChunkCellList (Section 3.1), i.e. its CellID is changed back to 0xFFFF, and its Status is changed to UNSCHEDULED.

## 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

- [RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.
- [I-D.ietf-6tisch-tsch]  
Watteyne, T., Palattella, M., and L. Grieco, "Using IEEE802.15.4e TSCH in an LLN context: Overview, Problem Statement and Goals", draft-ietf-6tisch-tsch-00 (work in progress), November 2013.
- [I-D.ietf-6tisch-architecture]  
Thubert, P., Watteyne, T., and R. Assimiti, "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-architecture-02 (work in progress), June 2014.
- [I-D.ietf-6tisch-terminology]  
Palattella, M., Thubert, P., Watteyne, T., and Q. Wang, "Terminology in IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-terminology-01 (work in progress), February 2014.
- [I-D.ietf-6tisch-minimal]  
Vilajosana, X. and K. Pister, "Minimal 6TiSCH Configuration", draft-ietf-6tisch-minimal-01 (work in progress), June 2014.
- [I-D.ietf-6tisch-6top-interface]  
Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top) Interface", draft-ietf-6tisch-6top-interface-00 (work in progress), March 2014.
- [I-D.wang-6tisch-6top-sublayer]  
Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top)", draft-wang-6tisch-6top-sublayer-00 (work in progress), February 2014.
- [I-D.ietf-6tisch-coap]  
Sudhaakar, R. and P. Zand, "6TiSCH Resource Management and Interaction using CoAP", draft-ietf-6tisch-coap-00 (work in progress), May 2014.

## 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.

[OpenWSN] 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 , August 2012.

[morell04label]

Morell, A., Vilajosana, X., Lopez-Vicario, J., and T. Watteyne, "Label Switching over IEEE802.15.4e Networks. Transactions on Emerging Telecommunications Technologies", June 2013.

## Authors' Addresses

Qin Wang (editor)  
Univ. of Sci. and Tech. Beijing  
30 Xueyuan Road  
Beijing, Hebei 100083  
China

Phone: +86 (10) 6233 4781  
Email: wangqin@ies.ustb.edu.cn

Xavier Vilajosana  
Universitat Oberta de Catalunya  
156 Rambla Poblenou  
Barcelona, Catalonia 08018  
Spain

Phone: +34 (646) 633 681  
Email: xvilajosana@uoc.edu

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

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



6TiSCH  
Internet-Draft  
Intended status: Standards Track  
Expires: January 5, 2015

P. Thubert, Ed.  
Cisco  
T. Watteyne  
Linear Technology  
RA. Assimiti  
Centero  
July 4, 2014

An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4e  
draft-ietf-6tisch-architecture-03

Abstract

This document presents an architecture for an IPv6 Multi-Link 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. The TSCH schedule can be static or dynamic. 6TiSCH defines mechanisms to establish and maintain the routing and scheduling operations in a centralized, distributed, or mixed fashion. Backbone Routers perform proxy Neighbor Discovery operations over the backbone on behalf of the wireless devices, 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 5, 2015.

## Copyright Notice

Copyright (c) 2014 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 . . . . .	3
2. Terminology . . . . .	4
3. Applications and Goals . . . . .	4
4. Overview and Scope . . . . .	5
5. Communication Paradigms and Interaction Models . . . . .	9
6. TSCH and 6top . . . . .	9
6.1. 6top . . . . .	9
6.2. 6top and RPL Objective Function operations . . . . .	10
6.3. Network Synchronization . . . . .	11
6.4. SlotFrames and Priorities . . . . .	12
6.5. Distributing the reservation of cells . . . . .	14
7. Schedule Management Mechanisms . . . . .	15
7.1. Minimal Static Scheduling . . . . .	15
7.2. Neighbor-to-neighbor Scheduling . . . . .	16
7.3. Remote Monitoring and Schedule Management . . . . .	17
7.4. Hop-by-hop Scheduling . . . . .	17
8. Forwarding Models . . . . .	18
8.1. Track Forwarding . . . . .	18
8.1.1. Transport Mode . . . . .	19
8.1.2. Tunnel Mode . . . . .	20
8.1.3. Tunnel Metadata . . . . .	21
8.2. Fragment Forwarding . . . . .	22
8.3. IPv6 Forwarding . . . . .	23
9. Centralized vs. Distributed Routing . . . . .	24
9.1. Packet Marking and Handling . . . . .	24
10. IANA Considerations . . . . .	25
11. Security Considerations . . . . .	25
12. Contributors . . . . .	25
13. Acknowledgements . . . . .	25
14. References . . . . .	25
14.1. Normative References . . . . .	26

14.2. Informative References . . . . .	27
14.3. External Informative References . . . . .	29
Authors' Addresses . . . . .	29

## 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 would 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 crosses the network from node to node following a very precise schedule, as a train that leaves intermediate stations at precise times along its path. With TSCH, time is formatted into timeSlots, and an individual cell is allocated to unicast or broadcast communication at the MAC level. The time slotted operation reduces collisions, saves energy, and enables to more closely engineer the network for deterministic properties. The channel hopping aspect is a simple and efficient technique to combat multipath fading and external interference (for example by WiFi emitters).

This document presents an architecture for an IPv6 Multi-Link 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. timeSlots and other device resources are managed by an abstract Network Management

Entity (NME) that may cooperate with the PCE in order to minimize the interaction with and the load on the constrained device.

## 2. Terminology

Readers are expected to be familiar with all the terms and concepts that are discussed in "neighbor Discovery for IP version 6" [RFC4861], "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals" [RFC4919], neighbor Discovery Optimization for Low-power and Lossy Networks [RFC6775] and "Multi-link Subnet Support in IPv6" [I-D.ietf-ipv6-multilink-subnets].

Readers may benefit from reading the "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks" [RFC6550] specification; "Multi-Link Subnet Issues" [RFC4903]; "Mobility Support in IPv6" [RFC6275]; "neighbor Discovery Proxies (ND Proxy)" [RFC4389]; "IPv6 Stateless Address Autoconfiguration" [RFC4862]; "FCFS SAVI: First-Come, First-Served Source Address Validation Improvement for Locally Assigned IPv6 Addresses" [RFC6620]; and "Optimistic Duplicate Address Detection" [RFC4429] prior to this specification for a clear understanding of the art in ND-proxying and binding.

The draft uses terminology defined or referenced in [I-D.ietf-6tisch-terminology], [I-D.chakrabarti-nordmark-6man-efficient-nd], [I-D.ietf-roll-rpl-industrial-applicability], [RFC5191] and [RFC4080].

The draft also conforms to the terms and models described in [RFC3444] and [RFC5889] and uses the vocabulary and the concepts defined in [RFC4291] for the IPv6 Architecture.

## 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 [IEEE802154e] and the centralized PCE. 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.

An incremental set of industrial requirements are addressed with the addition of an autonomic and distributed routing operation 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-vehicle 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.ietf-6tisch-tsch] MAC Low Power Lossy Network (LLN).

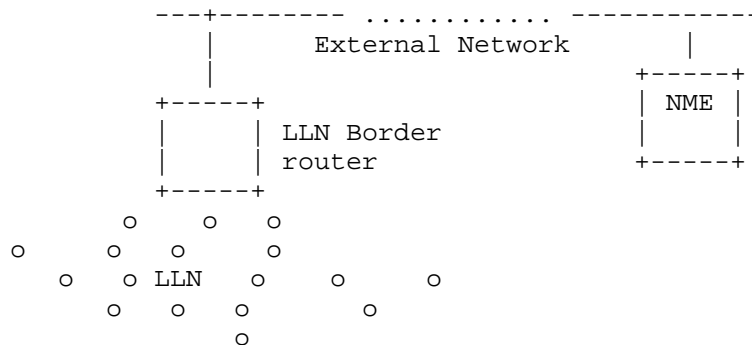


Figure 1: Basic Configuration

The LLN devices communicate over IPv6 [RFC2460] using the 6LoWPAN Header Compression (6LoWPAN HC) [RFC6282]. From the perspective of Layer 3, a single LLN interface (typically an IEEE802.15.4-compliant radio) may be seen as a collection of Links with different capabilities for unicast or multicast services. An IPv6 subnet spans over multiple links, effectively forming a Multi-Link subnet. Within that subnet, neighbor Devices are discovered with 6LoWPAN neighbor Discovery (6LoWPAN ND) [RFC6775]. RPL [RFC6550] enables routing within the LLN, typically within the Multi-Link subnet in the so called Route 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, the LLN Border Router (LBR), acts as RPL root, 6LoWPAN HC terminator, and LLN Border Router (LBR) to the outside. The LBR is usually powered. More on RPL Instances can be found in RPL [RFC6550], sections "3.1.2. RPL Identifiers" and "3.1.3. Instances, DODAGs, and DODAG Versions".

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 can be handled by the Protocol for Carrying Authentication for Network access (PANA) [RFC5191].

The LLN devices are time-synchronized at the MAC level. The LBR that serves as time source is a RPL parent in a particular RPL Instance that serves for time synchronization; this way, the time synchronization starts at the RPL root and follows the RPL DODAGs with no timing loop.

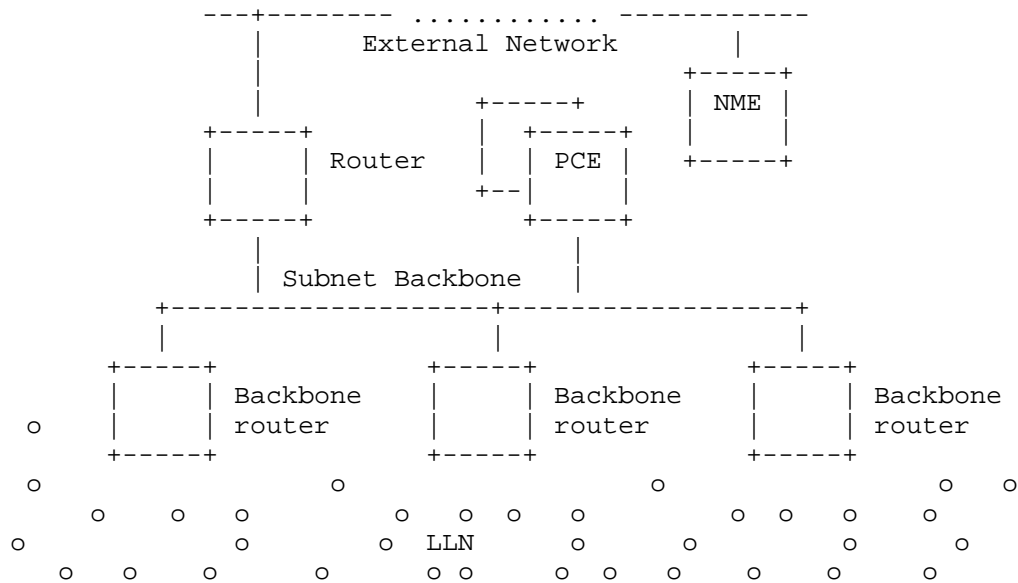


Figure 2: Extended Configuration

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. 6TiSCH 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:

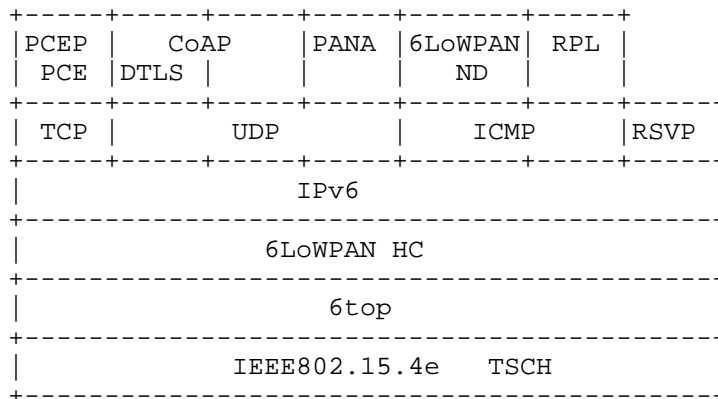


Figure 3: 6TiSCH stack

RPL is the routing protocol of choice for LLNs. (TBD RPL) whether there is a need to define a 6TiSCH 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 Block and Observe patterns), SenML and CoAP Resource Directory.

(tbd PCE) 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 PANA) There is a debate whether PANA (layer 3) or IEEE802.1x (layer 2) should be used in the join process. There is also a debate whether the node should be able to send any unprotected packet on the medium. Regardless, the security model must ensure that, prior to a join process, packets from a untrusted device must be controlled in volume and in reachability.

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

IEEE802.1TSN: external, maintain consistency. See also AVnu.

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

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



The 6TiSCH Operation sublayer (6top) [I-D.wang-6tisch-6top-sublayer] is an Logical Link Control (LLC) or a portion thereof that provides the abstraction of an IP link over a TSCH MAC.

## 5. Communication Paradigms and Interaction Models

[I-D.ietf-6tisch-terminology] defines the terms of Communication Paradigms and Interaction Models, which can be placed in parallel to the Information Models and Data Models that are defined in [RFC3444].

A Communication Paradigms would be an abstract view of a protocol exchange, and would come with an Information Model for the information that is being exchanged. In contrast, an Interaction Models would be more refined and could point on standard operation such as a Representational state transfer (REST) "GET" operation and would match a Data Model for the data that is provided over the protocol exchange.

[I-D.ietf-roll-rpl-industrial-applicability] section 2.1.3. and next discusses application-layer paradigms, such as Source-sink (SS) that is a Multipeer to Multipeer (MP2MP) model that is primarily used for alarms and alerts, Publish-subscribe (PS, or pub/sub) that is typically used for sensor data, as well as Peer-to-peer (P2P) and Peer-to-multipeer (P2MP) communications. Additional considerations on Duocast and its N-cast generalization are also provided. Those paradigms are frequently used in industrial automation, which is a major use case for IEEE802.15.4e TSCH wireless networks with [ISA100.11a] and [WirelessHART], that provides a wireless access to [HART] applications and devices.

This specification focuses on Communication Paradigms and Interaction Models for packet forwarding and TSCH resources (cells) management. Management mechanisms for the TSCH schedule at Link-layer (one-hop), Network-layer (multithop along a track), and Application-layer (remote control) are discussed in Section 7. Link-layer frame forwarding interactions are discussed in Section 8, and Network-layer Packet routing is addressed in section Section 9.

## 6. TSCH and 6top

### 6.1. 6top

6top is a logical link control sitting between the IP layer and the TSCH MAC layer, which provides the link abstraction that is required for IP operations. The 6top operations are specified in [I-D.wang-6tisch-6top-sublayer]. In particular, 6top provides a management interface that enables an external management entity to schedule cells and slotFrames, and allows the addition of

complementary functionality, for instance to support a dynamic schedule management based on observed resource usage as discussed in section Section 7.2. The 6top data model and management interfaces are further discussed in Section 7.3.

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 for example 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 exact 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.

## 6.2. 6top and RPL Objective Function operations

An implementation of a RPL [RFC6550] Objective Function (OF), such as the RPL Objective Function Zero (OF0) [RFC6552] that is used in the Minimal 6TiSCH Configuration [I-D.ietf-6tisch-minimal] to support RPL over a static schedule, may leverage, for its internal computation, the information maintained by 6top.

In particular, 6top creates and maintains an abstract neighbor table. A neighbor table entry contains a set of statistics with respect to that specific neighbor including the time when the last packet has been received from that neighbor, a set of cell quality metrics (RSSI, LQI), the number of packets sent to the neighbor or the number of packets received from it. This information can be obtained through 6top management APIs as detailed in the 6top sublayer specification [I-D.wang-6tisch-6top-sublayer] and used to compute a Rank Increment that will determine the selection of the preferred parent.

6top provides statistics about the underlying layer so the OF can be tuned to the nature of the TSCH MAC layer. 6top also enables the RPL OF to influence the MAC behaviour, for instance by configuring the periodicity of IEEE802.15.4e Extended Beacons (EB's). By augmenting the EB periodicity, it is possible to change the network dynamics so as to improve the support of devices that may change their point of attachment in the 6TiSCH network.

Some RPL control messages, such as the DODAG Information Object (DIO) are ICMPv6 messages that are broadcast to all neighbor nodes. With

6TiSCH, the broadcast channel requirement is addressed by 6top by configuring TSCH to provide a broadcast channel, as opposed to, for instance, piggybacking the DIO messages in Enhance Beacons.

In the TSCH schedule, each cell has the IEEE802.15.4e LinkType attribute. Setting the LinkType to ADVERTISING indicates that the cell MAY be used to send an Enhanced Beacon. When a node forms its Enhanced Beacon, the cell, with LinkType=ADVERTISING, SHOULD be included in the FrameAndLinkIE, and its LinkOption field SHOULD be set to the combination of "Receive" and "Timekeeping". The receiver of the Enhanced Beacon MAY be listening at the cell to get the Enhanced Beacon ([IEEE802154e]). 6top takes this way to establish broadcast channel, which not only allows TSCH to broadcast Enhanced Beacons, but also allows an upper layer like RPL.

To broadcast ICMPv6 control messages used by RPL such as DIO or DAO, 6top uses the payload of a Data frames. The message is inserted into the queue associated with the cells which LinkType is set to ADVERTISING. Then, taking advantage of the broadcast cell feature established with FrameAndLinkIE (as described above), the RPL control message can be received by neighbors, which enables the maintenance of RPL DODAGs.

A LinkOption combining "Receive" and "Timekeeping" bits indicates to the receivers of the Enhanced Beacon that the cell MUST be used as a broadcast cell. The frequency of sending Enhanced Beacons or other broadcast messages by the upper layer is determined by the timers associated with the messages. For example, the transmission of Enhance Beacons is triggered by a timer in 6top; transmission of a DIO message is triggered by the trickle timer of RPL.

### 6.3. Network Synchronization

Nodes in a TSCH network must be time synchronized. A node keeps synchronized to its time source neighbor through a combination of frame-based and acknowledgement-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 synchronization 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 a management entity.

Time distribution requires a loop-less structure. Nodes taken in a synchronization loop will rapidly desynchronize from the network and become isolated. It is expected that a RPL DAG with a dedicated global Instance is deployed for the purpose of time synchronization.

That Instance is referred to as the Time Synchronization Global Instance (TSGI). The TSGI can be operated in either of the 3 modes that are detailed in RPL [RFC6550] section "3.1.3. Instances, DODAGs, and DODAG Versions". Multiple uncoordinated DODAGs with independent roots may be used if all the roots share a common time source such as the Global Positioning System (GPS). In the absence of a common time source, the TSGI should form a single DODAG with a virtual root. A backbone network is then used to synchronize and coordinate RPL operations between the backbone routers that act as sinks for the LLN.

A node that has not joined the TSGI advertises a MAC level Join Priority of 0xFF to notify its neighbors that it is not capable of serving as time parent. A node that has joined the TSGI advertises a MAC level Join Priority set to its DAGRank() in that Instance, where DAGRank() is the operation specified in [RFC6550], section "3.5.1. Rank Comparison".

A root is configured or obtains by some external means the knowledge of the RPLInstanceID for the TSGI. The root advertises its DagRank in the TSGI, that MUST be less than 0xFF, as its Join Priority (JP) in its IEEE802.15.4e Extended Beacons (EB). We'll note that the JP is now specified between 0 and 0x3F leaving 2 bits in the octet unused in the IEEE802.15.4e specification. After consultation with IEEE authors, it was asserted that 6TiSCH can make a full use of the octet to carry an integer value up to 0xFF.

A node that reads a Join Priority of less than 0xFF should join the neighbor with the lesser Join Priority and use it as time parent. If the node is configured to serve as time parent, then the node should join the TSGI, obtain a Rank in that Instance and start advertising its own DagRank in the TSGI as its Join Priority in its EBs.

#### 6.4. SlotFrames and Priorities

6TiSCH enables in essence the capability to use IPv6 over a MAC layer that enables to schedule some of the transmissions. In order to ensure that the medium is free of contending packets when time arrives for a scheduled transmission, a window of time is defined around the scheduled transmission time where the medium must be free of contending energy.

One simple way to obtain such a window is to format time and frequencies in cells of transmission of equal duration. This is the method that is adopted in IEEE802.15.4e TSCH as well as the Long Term Evolution (LTE) of cellular networks.

In order to describe that formatting of time and frequencies, the 6TiSCH architecture defines a global concept that is called a Channel Distribution and Usage (CDU) matrix; a CDU matrix is a matrix of cells with an height equal to the number of available channels (indexed by ChannelOffsets), a timeSlot duration (10-15 milliseconds are typical in 802.15.4e TSCH) and a width (in timeSlots) that is the period of the network scheduling operation (indexed by slotOffsets) for that CDU matrix.

A CDU matrix iterates over and over with a pseudo-random rotation from an epoch time. In a given network, there might be multiple CDU matrices that operate with different width, so they have different durations and represent different periodic operations. It is RECOMMENDED that all CDU matrices in a 6TiSCH domain operate with the same cell duration and are aligned, so as to optimize the chances of interferences from slotted-aloha operations. The knowledge of the CDU matrices is shared between all the nodes and used in particular to define slotFrames.

A slotFrame is a MAC-level abstraction that is common to all nodes and contains a series of timeSlots of equal length and precedence. It is characterized by a slotFrame\_ID, and a slotFrame\_size. A slotFrame aligns to a CDU matrix for its parameters, such as number and duration of timeSlots.

Multiple slotFrames can coexist in a node schedule, i.e., a node can have multiple activities scheduled in different slotFrames, based on the precedence of the 6TiSCH topologies. The slotFrames may be aligned to different CDU matrices and thus have different width. There is typically one slotFrame for scheduled traffic that has the highest precedence and one or more slotFrame(s) for RPL traffic. The timeSlots in the slotFrame are indexed by the SlotOffset; the first cell is at SlotOffset 0.

A 6TiSCH Instance is associated to one slotFrame. A slotFrame may be shared by multiple Instances of equal relative precedence. Within an Instance, 6top uses priority queues to manage concurrent data flows of different priorities within an Instance and between Instances of a same precedence, associated to a given IPv6 link and a given bundle of TX-cells. When a packet is received from an higher layer for transmission, 6top inserts that packet in the outgoing queue which matches the packet best (DSCP can therefore be used). At each scheduled transmit slot, 6top looks for the frame in all the outgoing queues that best matches the cells. If a frame is found, it is given to the TSCH MAC for transmission.

## 6.5. Distributing the reservation of cells

6TiSCH expects a high degree of scalability together with a distributed routing functionality based on RPL. To achieve this goal, the spectrum must be allocated in a way that allows for spatial reuse between zones that will not interfere with one another. In a large and spatially distributed network, a 6TiSCH node is often in a good position to determine usage of spectrum in its vicinity.

Use cases for distributed routing are often associated with a statistical distribution of best-effort traffic with variable needs for bandwidth on each individual link. With 6TiSCH, the link abstraction is implemented as a bundle of cells; the size of a bundle is optimal when both the energy wasted idle listening and the packet drops due to congestion loss are minimized. This can be maintained if the number of cells in a bundle is adapted dynamically, and with enough reactivity, to match the variations of best-effort traffic. In turn, the agility to fulfil the needs for additional cells improves when the number of interactions with other devices and the protocol latencies are minimized.

6TiSCH limits that interaction to RPL parents that will only negotiate with other RPL parents, and performs that negotiation by groups of cells as opposed to individual cells. The 6TiSCH architecture allows RPL parents to adjust dynamically, and independently from the PCE, the amount of bandwidth that is used to communicate between themselves and their children, in both directions; to that effect, an allocation mechanism enables a RPL parent to obtain the exclusive use of a portion of a CDU matrix within its interference domain.

The 6TiSCH architecture introduces the concept of chunks [I-D.ietf-6tisch-terminology]) to operate such spectrum distribution for a whole group of cells at a time. The CDU matrix is formatted into a set of chunks, each of them identified uniquely by a chunk-ID. The knowledge of this formatting is shared between all the nodes in a 6TiSCH network. 6TiSCH also defines the process of chunk ownership appropriation whereby a RPL parent discovers a chunk that is not used in its interference domain (e.g lack of energy detected in reference cells in that chunk); then claims the chunk, and then defends it in case another RPL parent would attempt to appropriate it while it is in use. The chunks is the basic unit of ownership that is used in that process.

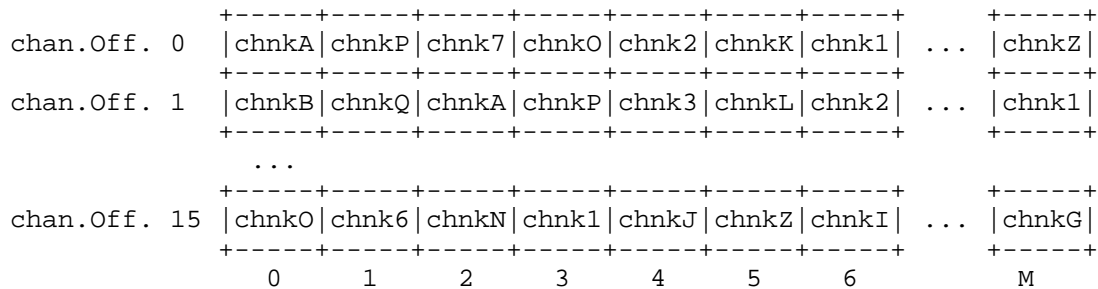


Figure 4: CDU matrix Partitioning in Chunks

As a result of the process of chunk ownership appropriation, the RPL parent has exclusive authority to decide which cell in the appropriated chunk can be used by which node in its interference domain. In other words, it is implicitly delegated the right to manage the portion of the CDU matrix that is represented by the chunk. The RPL parent may thus orchestrate which transmissions occur in any of the cells in the chunk, by allocating cells from the chunk to any form of communication (unicast, multicast) in any direction between itself and its children. Initially, those cells are added to the heap of free cells, then dynamically placed into existing bundles, in new bundles, or allocated opportunistically for one transmission.

The appropriation of a chunk can also be requested explicitly by the PCE to any node. In that case, the node still may need to perform the appropriation process to validate that no other node has claimed that chunk already. After a successful appropriation, the PCE owns the cells in that chunk, and may use them as hard cells to set up tracks.

## 7. Schedule Management Mechanisms

6TiSCH uses 4 paradigms to manage the TSCH schedule of the LLN nodes: Static Scheduling, neighbor-to-neighbor Scheduling, remote monitoring and scheduling management, and Hop-by-hop scheduling. Multiple mechanisms are defined that implement the associated Interaction Models, and can be combined and used in the same LLN. Which mechanism(s) to use depends on application requirements.

### 7.1. Minimal Static Scheduling

In the simplest instantiation of a 6TiSCH network, a common fixed schedule may be shared by all nodes in the network. Cells are shared, and nodes contend for slot access in a slotted aloha manner.

A static TSCH schedule can be used to bootstrap a network, as an initial phase during implementation, or as a fall-back mechanism in case of network malfunction. This scheduled can be preconfigured or learnt by a node when joining the network. Regardless, the schedule remains unchanged after the node has joined a network. The Routing Protocol for LLNs (RPL) is used on the resulting network. This "minimal" scheduling mechanism that implements this paradigm is detailed in [I-D.ietf-6tisch-minimal].

## 7.2. Neighbor-to-neighbor Scheduling

In the simplest instantiation of a 6TiSCH network described in Section 7.1, nodes may expect a packet at any cell in the schedule and will waste energy idle listening. In a more complex instantiation of a 6TiSCH network, a matching portion of the schedule is established between peers to reflect the observed amount of transmissions between those nodes. The aggregation of the cells between a node and a peer forms a bundle that the 6top layer uses to implement the abstraction of a link for IP. The bandwidth on that link is proportional to the number of cells in the bundle.

If the size of a bundle is configured to fit an average amount of bandwidth, peak emissions will be destroyed. If the size is configured to allow for peak emissions, energy is be wasted idle listening.

In the most efficient instantiation of a 6TiSCH network, the size of the bundles that implement the links may be changed dynamically in order to adapt to the need of end-to-end flows routed by RPL. An optional On-The-Fly (OTF) component may be used to monitor bandwidth usage and perform requests for dynamic allocation by the 6top sublayer. The OTF component is not part of the 6top sublayer. It may be collocated on the same device or may be partially or fully offloaded to an external system.

The 6top sublayer [I-D.wang-6tisch-6top-sublayer] defines a protocol for neighbor nodes to reserve soft cells to one another. Because this reservation is done without global knowledge of the schedule of nodes in the LLN, scheduling collisions are possible. 6top defines a monitoring process which continuously tracks the packet delivery ratio of soft cells. It uses these statistics to trigger the relocation of a soft cell in the schedule, using a negotiation protocol between the neighbors nodes communicating over that cell.

Monitoring and relocation is done in the 6top layer. For the upper layer, the connection between two neighbor nodes appears as an number of cells. Depending on traffic requirements, the upper layer can request 6top to add or delete a number of cells scheduled to a



particular neighbor, without being responsible for choosing the exact slotOffset/channelOffset of those cells.

### 7.3. Remote Monitoring and Schedule Management

The 6top interface document [I-D.ietf-6tisch-6top-interface] specifies the generic data model that can be used to monitor and manage resources at the 6top sublayer. Abstract methods are suggested for use by a management entity in the device. The data model also enables remote control operations on the 6top sublayer.

Being able to interact with the 6top sublayer of a node multiple hops away can be used for monitoring, scheduling, or a combination of both. The architecture supports variations on the deployment model, and focuses on the flows rather than whether there is a proxy or a translational operation on the way.

[I-D.ietf-6tisch-coap] defines an mapping of 6top's set of commands described in [I-D.ietf-6tisch-6top-interface] to CoAP resources. This allows an entity to interact with the 6top layer of a node that is multiple hops away in a RESTful fashion.

[I-D.ietf-6tisch-coap] defines a basic set CoAP resources and associated RESTful access methods (GET/PUT/POST/DELETE). The payload (body) of the CoAP messages is encoded using the CBOR format. The draft also defines the concept of "profiles" to allow for future or specific extensions, as well as a mechanism for a CoAP client to discover the profiles installed on a node.

The entity issuing the CoAP requests can be a central scheduling entity (e.g. a PCE), a node multiple hops away with the authority to modify the TSCH schedule (e.g. the head of a local cluster), or a external device monitoring the overall state of the network (e.g. NME). The architecture allows for different types of interactions between this CoAP client and a node in the network:

### 7.4. Hop-by-hop Scheduling

A node can reserve a track to a destination node multiple hops away by installing soft cells at each intermediate node. This forms a track of soft cells. It is the responsibility of the 6top sublayer of each node on the track to monitor these soft cells and trigger relocation when needed.

This hop-by-hop reservation mechanism is similar to [RFC2119] and [RFC5974]. The protocol for a node to trigger hop-by-hop scheduling is not yet defined.

## 8. Forwarding Models

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

### 8.1. Track Forwarding

Track Forwarding is the simplest and fastest. A bundle of cells set to receive (RX-cells) is uniquely paired to a bundle of cells that are set to transmit (TX-cells), representing a layer-2 forwarding state that can be used regardless of the network layer protocol. This model can effectively be seen as a Generalized Multi-protocol Label Switching (G-MPLS) operation in that the information used to switch a frame is not an explicit label, but rather related to other properties of the way the packet was received, a particular cell in the case of 6TiSCH. 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 data frame that is forwarded along a Track normally has a destination MAC address that is set to broadcast - or a multicast address depending on MAC support. This way, the MAC layer in the intermediate nodes accepts the incoming frame and 6top switches it without incurring a change in the MAC header. In the case of IEEE802.15.4e, this means effectively broadcast, so that along the Track the short address for the destination of the frame is set to 0xFFFF.

A Track is thus formed end-to-end as a succession of paired bundles, a receive bundle from the previous hop and a transmit bundle to the next hop along the Track, and a cell in such a bundle belongs to at most one Track. For a given iteration of the device schedule, the effective channel of the cell is obtained by adding a pseudo-random number to the channelOffset of the cell, which results in a rotation of the frequency that used for transmission. The bundles may be computed so as to accommodate both variable rates and retransmissions, so they might not be fully used at a given iteration of the schedule. The 6TiSCH architecture provides additional means to avoid waste of cells as well as overflows in the transmit bundle, as follows:

In one hand, a TX-cell that is not needed for the current iteration may be reused opportunistically on a per-hop basis for routed packets. When all of the frame that were received for a given Track are effectively transmitted, any available TX-cell for that Track can

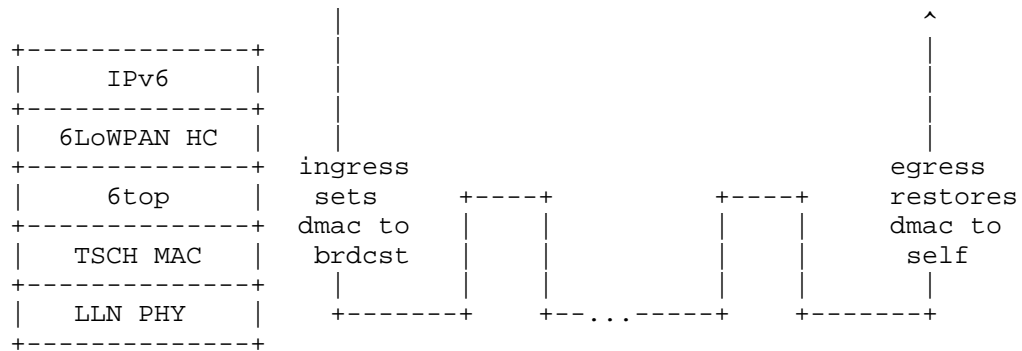
be reused for upper layer traffic for which the next-hop router matches the next hop along the Track. In that case, the cell that is being used is effectively a TX-cell from the Track, but the short address for the destination is that of the next-hop router. It results that a frame that is received in a RX-cell of a Track with a destination MAC address set to this node as opposed to broadcast must be extracted from the Track and delivered to the upper layer (a frame with an unrecognized MAC address is dropped at the lower MAC layer and thus is not received at the 6top sublayer).

On the other hand, it might happen that there are not enough TX-cells in the transmit bundle to accommodate the Track traffic, for instance if more retransmissions are needed than provisioned. In that case, the frame can be placed for transmission in the bundle that is used for layer-3 traffic towards the next hop along the track as long as it can be routed by the upper layer, that is, typically, if the frame transports an IPv6 packet. The MAC address should be set to the next-hop MAC address to avoid confusion. It results that a frame that is received over a layer-3 bundle may be in fact associated to a Track. In a classical IP link such as an Ethernet, off-track traffic is typically in excess over reservation to be routed along the non-reserved path based on its QoS setting. But with 6TiSCH, since the use of the layer-3 bundle may be due to transmission failures, it makes sense for the receiver to recognize a frame that should be re-tracked, and to place it back on the appropriate bundle if possible. A frame should be re-tracked if the Per-Hop-Behavior group indicated in the Differentiated Services Field in the IPv6 header is set to Deterministic Forwarding, as discussed in Section 9.1. A frame is re-tracked by scheduling it for transmission over the transmit bundle associated to the Track, with the destination MAC address set to broadcast.

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

#### 8.1.1.1. Transport Mode

In transport mode, the Protocol Data Unit (PDU) is associated with flow-dependant meta-data that refers uniquely to the Track, so the 6top sublayer can place the frame in the appropriate cell without ambiguity. In the case of IPv6 traffic, this flow identification is transported in the Flow Label of 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 destination MAC address (DMAC) and punting to the upper layer.



Track Forwarding, Transport Mode

#### 8.1.2. Tunnel Mode

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

In that mode, some entity (e.g. PCE) can coordinate with a WirelessHART Network Manager or an ISA100.11a System Manager to specify the flows that are to be transported transparently over the Track.

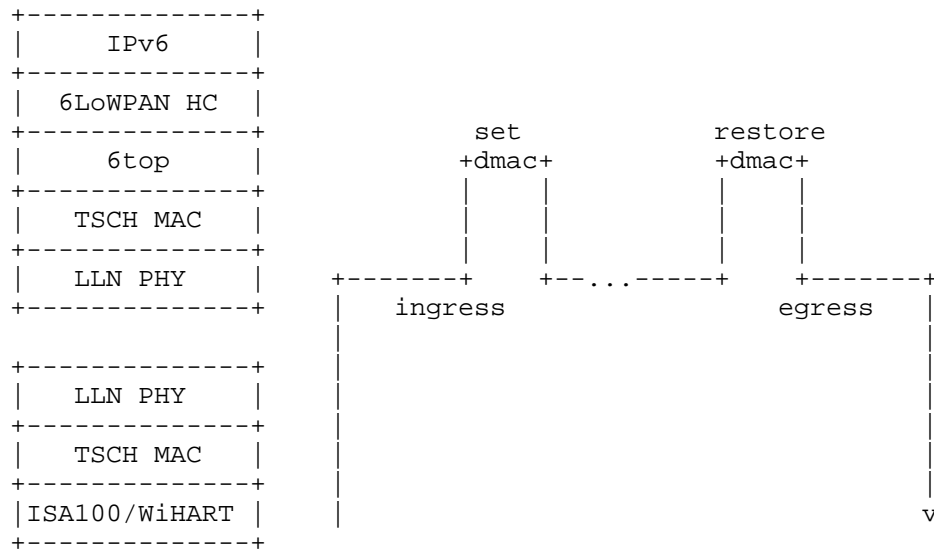


Figure 5: Track Forwarding, Tunnel Mode

In that case, the flow information that identifies the Track at the ingress 6TiSCH router is derived from the RX-cell. The dmac is set to this node but the flow information indicates that the frame must be tunnelled over a particular Track so the frame is not passed to the upper layer. Instead, the dmac is forced to broadcast and the frame is passed to the 6top sublayer for switching.

At the egress 6TiSCH 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.

#### 8.1.3. 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.

## 8.2. Fragment Forwarding

Considering that 6LoWPAN packets can be as large as 1280 bytes (the IPv6 MTU), and that the non-storing mode of RPL implies Source Routing that requires space for routing headers, and that a IEEE802.15.4 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 6TiSCH network, Hop-by-Hop recomposition occurs 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 undetermined 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.

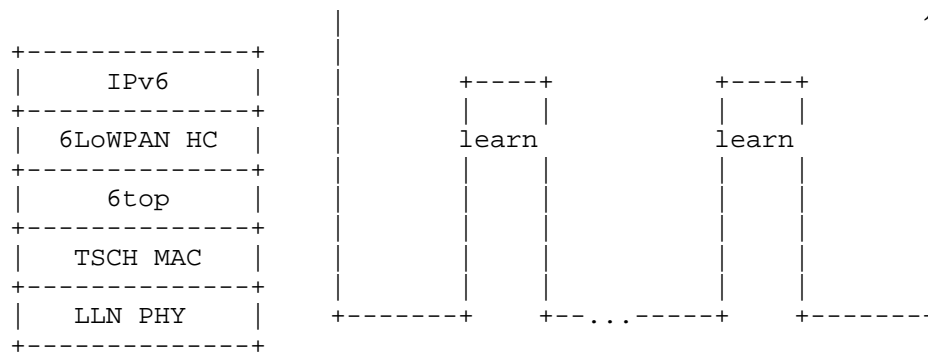


Figure 6: Forwarding First Fragment

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.

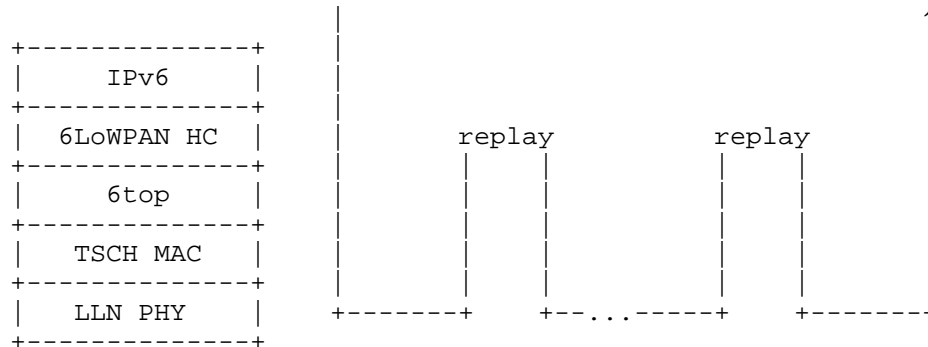


Figure 7: Forwarding Next Fragment

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.

### 8.3. IPv6 Forwarding

As the packets are routed at layer 3, traditional QoS and RED operations are expected to prioritize flows; the application of Differentiated Services is further discussed in [I-D.svshah-tsvwg-lln-diffserv-recommendations].

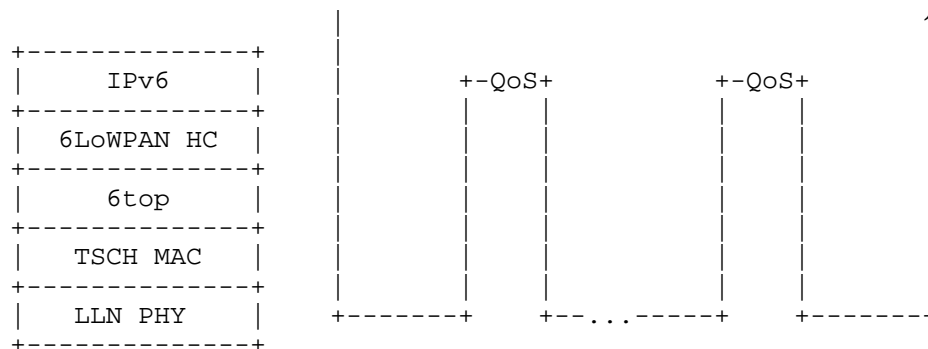


Figure 8: IP Forwarding

## 9. Centralized vs. Distributed Routing

6TiSCH supports a mixed model of centralized routes and distributed routes. Centralized routes can for example be computed by an entity such as a PCE. Distributed routes are computed by RPL.

Both methods may inject routes in the Routing Tables of the 6TiSCH routers. In either case, each route is associated with a 6TiSCH topology that can be a RPL Instance topology or a track. The 6TiSCH topology is indexed by an Instance ID, in a format that reuses the RPLInstanceID as defined in RPL [RFC6550].

Both RPL and PCE rely on shared sources such as policies to define Global and Local RPLInstanceIDs that can be used by either method. It is possible for centralized and distributed routing to share a same topology. Generally they will operate in different slotFrames, and centralized routes will be used for scheduled traffic and will have precedence over distributed routes in case of conflict between the slotFrames.

### 9.1. Packet Marking and Handling

All packets inside a 6TiSCH domain MUST carry the Instance ID that identifies the 6TiSCH topology that is to be used for routing and forwarding that packet. The location of that information MUST be the same for all packets forwarded inside the domain.

For packets that are routed by a PCE along a Track, the tuple formed by the IPv6 source address and a local RPLInstanceID in the packet identify uniquely the Track and associated transmit bundle. Additionally, an IP packet that is sent along a Track uses the Differentiated Services Per-Hop-Behavior Group called Deterministic Forwarding, as described in [I-D.svshah-tsvwg-deterministic-forwarding].

For packets that are routed by RPL [RFC6550], that information is the RPLInstanceID that is carried as part of the RPL Packet Information, which is defined in section 11.2 "Loop Avoidance and Detection".

At the time of this writing, there are 2 methods to transport the RPL Packet Information in an IPv6 packet, either in a IPv6 Hop-By-Hop Header, or encoded in a compressed fashion in the IPv6 Flow Label.

The former method places a RPL option [RFC6553] in the IPv6 Hop-By-Hop Header. It MUST be used if at least one RPL Instance uses a MinHopRankIncrease that is less than DEFAULT\_MIN\_HOP\_RANK\_INCREASE (defined to 256 in [RFC6550]), which bars the capability to compress the SenderRank in the RPL Packet Information to a single octet. If



that is not the case, it is RECOMMENDED to use the latter method of encoding the RPL Packet Information in the Flow Label, which is specified in [I-D.thubert-6man-flow-label-for-rpl].

Either way, the method and format used for encoding the RPLInstanceID is generalized to all 6TiSCH topological Instances, which include both RPL Instances and Tracks.

#### 10. IANA Considerations

This specification does not require IANA action.

#### 11. Security Considerations

This specification is not found to introduce new security threat.

#### 12. Contributors

The editors and authors wish to recognize the contribution of

Xavier Vilajosana who lead the design of the minimal support with RPL and contributed deeply to the 6top design.

Qin Wang who lead the design of the 6top sublayer and contributed related text that was moved and/or adapted in this document.

#### 13. Acknowledgements

This specification is the result interactions in particular during the 6TiSCH (bi)Weekly call. The authors wish to thank: Alaeddine Weslati, Alfredo Grieco, Bert Greevenbosch, Cedric Adjih, Diego Dujovne, Dominique Barthel, Elvis Vogli, Geraldine Texier, Giuseppe Piro, Guillaume Gaillard, Herman Storey, Ines Robles, Jonathan Simon, Kazushi Muraoka, Ken Bannister, Kuor Hsin Chang, Laurent Toutain, Maik Seewald, Maria Rita Palattella, Michael Behringer, Michael Richardson, Nancy Cam Winget, Nicola Accettura, Nicolas Montavont, Oleg Hahm, Pat Kinney, Patrick Wetterwald, Paul Duffy, Peter van der Stock, Pieter de Mil, Pouria Zand, Rouhollah Nabati, Rafa Marin-Lopez, Raghuram Sudhaakar, Rene Struik, Sedat Gormus, Shitanshu Shah, Steve Simlo, Subir Das, Tengfei Chang, Tina Tsou, Tom Phinney, Xavier Lagrange and Yoshihiro Ohba for their various participation.

#### 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. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", RFC 3444, January 2003.
- [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.
- [RFC4389] Thaler, D., Talwar, M., and C. Patel, "Neighbor Discovery Proxies (ND Proxy)", RFC 4389, April 2006.
- [RFC4429] Moore, N., "Optimistic Duplicate Address Detection (DAD) for IPv6", RFC 4429, April 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.
- [RFC4903] Thaler, D., "Multi-Link Subnet Issues", RFC 4903, June 2007.
- [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.
- [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.
- [RFC6275] Perkins, C., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, July 2011.
- [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.
- [RFC6552] Thubert, P., "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6552, March 2012.
- [RFC6553] Hui, J. and JP. Vasseur, "The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams", RFC 6553, March 2012.
- [RFC6620] Nordmark, E., Bagnulo, M., and E. Levy-Abegnoli, "FCFS SAVI: First-Come, First-Served Source Address Validation Improvement for Locally Assigned IPv6 Addresses", RFC 6620, May 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., Thubert, P., and M. Wasserman, "IPv6 Neighbor Discovery Optimizations for Wired and Wireless Networks", draft-chakrabarti-nordmark-6man-efficient-nd-05 (work in progress), February 2014.
- [I-D.ietf-6tisch-6top-interface]  
Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top) Interface", draft-ietf-6tisch-6top-interface-00 (work in progress), March 2014.

- [I-D.ietf-6tisch-coap]  
Sudhaakar, R. and P. Zand, "6TiSCH Resource Management and Interaction using CoAP", draft-ietf-6tisch-coap-00 (work in progress), May 2014.
- [I-D.ietf-6tisch-minimal]  
Vilajosana, X. and K. Pister, "Minimal 6TiSCH Configuration", draft-ietf-6tisch-minimal-01 (work in progress), June 2014.
- [I-D.ietf-6tisch-terminology]  
Palattella, M., Thubert, P., Watteyne, T., and Q. Wang, "Terminology in IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-terminology-01 (work in progress), February 2014.
- [I-D.ietf-6tisch-tsch]  
Watteyne, T., Palattella, M., and L. Grieco, "Using IEEE802.15.4e TSCH in an LLN context: Overview, Problem Statement and Goals", draft-ietf-6tisch-tsch-00 (work in progress), November 2013.
- [I-D.ietf-ipv6-multilink-subnets]  
Thaler, D. and C. Huitema, "Multi-link Subnet Support in IPv6", draft-ietf-ipv6-multilink-subnets-00 (work in progress), July 2002.
- [I-D.ietf-roll-rpl-industrial-applicability]  
Phinney, T., Thubert, P., and R. Assimiti, "RPL applicability in industrial networks", draft-ietf-roll-rpl-industrial-applicability-02 (work in progress), October 2013.
- [I-D.svshah-tsvwg-deterministic-forwarding]  
Shah, S. and P. Thubert, "Deterministic Forwarding PHB", draft-svshah-tsvwg-deterministic-forwarding-01 (work in progress), March 2014.
- [I-D.svshah-tsvwg-lln-diffserv-recommendations]  
Shah, S. and P. Thubert, "Differentiated Service Class Recommendations for LLN Traffic", draft-svshah-tsvwg-lln-diffserv-recommendations-02 (work in progress), February 2014.
- [I-D.thubert-6lowpan-backbone-router]  
Thubert, P., "6LoWPAN Backbone Router", draft-thubert-6lowpan-backbone-router-03 (work in progress), February 2013.

[I-D.thubert-6man-flow-label-for-rpl]

Thubert, P., "The IPv6 Flow Label within a RPL domain", draft-thubert-6man-flow-label-for-rpl-03 (work in progress), May 2014.

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

Thubert, P. and J. Hui, "LLN Fragment Forwarding and Recovery", draft-thubert-roll-forwarding-frags-02 (work in progress), September 2013.

[I-D.wang-6tisch-6top-sublayer]

Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top)", draft-wang-6tisch-6top-sublayer-00 (work in progress), February 2014.

#### 14.3. External Informative References

[HART] [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", .

[IEEE802.1TSNTG]

IEEE Standards Association, "IEEE 802.1 Time-Sensitive Networks Task Group", March 2013, <<http://www.ieee802.org/1/pages/avbridges.html>>.

[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.

[ISA100.11a]

ISA/ANSI, "Wireless Systems for Industrial Automation: Process Control and Related Applications - ISA100.11a-2011 - IEC 62734", 2011, <<http://www.isa.org/Community/SP100WirelessSystemsforAutomation>>.

[WirelessHART]

[www.hartcomm.org](http://www.hartcomm.org), "Industrial Communication Networks - Wireless Communication Network and Communication Profiles - WirelessHART - IEC 62591", 2010.

#### 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

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

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

Robert Assimiti  
Centero  
961 Indian Hills Parkway  
Marietta, GA 30068  
USA

Phone: +1 404 461 9614  
Email: robert.assimiti@centerotech.com

6TiSCH  
Internet-Draft  
Intended status: Informational  
Expires: January 5, 2015

X. Vilajosana, Ed.  
Universitat Oberta de Catalunya  
K. Pister  
University of California Berkeley  
July 4, 2014

Minimal 6TiSCH Configuration  
draft-ietf-6tisch-minimal-02

Abstract

This document describes the minimal set of rules to operate a [IEEE802154e] Timeslotted Channel Hopping (TSCH) network. This minimal mode of operation can be used during network bootstrap, as a fallback mode of operation when no dynamic scheduling solution is available or functioning, or during early interoperability testing and development.

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 5, 2015.

Copyright Notice

Copyright (c) 2014 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 . . . . .	3
2. Minimal Schedule Configuration . . . . .	3
2.1. Slotframe . . . . .	3
2.2. Cell Options . . . . .	5
2.3. Retransmissions . . . . .	6
2.4. Time Slot timing . . . . .	6
3. Enhanced Beacons Configuration and Content . . . . .	8
3.1. Sync IE . . . . .	8
3.1.1. IE Header . . . . .	8
3.1.2. IE Content . . . . .	8
3.2. TSCH Timeslot IE . . . . .	9
3.2.1. IE Header . . . . .	9
3.2.2. IE Content . . . . .	9
3.3. Channel Hopping IE . . . . .	9
3.3.1. IE Header . . . . .	9
3.3.2. IE Content . . . . .	9
3.4. Frame and Link IE . . . . .	9
3.4.1. IE Header . . . . .	9
3.4.2. IE Content . . . . .	10
4. Acknowledgment . . . . .	10
4.1. ACK/NACK Time Correction IE . . . . .	10
4.1.1. IE Header . . . . .	10
4.1.2. IE Content . . . . .	10
5. Neighbor information . . . . .	11
5.1. Neighbor Table . . . . .	11
5.2. Time Source Neighbor Selection . . . . .	12
6. Queues and Priorities . . . . .	12
7. RPL on TSCH . . . . .	13
7.1. RPL Objective Function Zero . . . . .	13
7.1.1. Rank computation . . . . .	13
7.1.2. Rank computation Example . . . . .	14
7.2. RPL Configuration . . . . .	16
7.2.1. Mode of Operation . . . . .	17
7.2.2. Trickle Timer . . . . .	17
7.2.3. Hysteresis . . . . .	17
7.2.4. Variable Values . . . . .	17



8. Acknowledgements . . . . .	18
9. References . . . . .	18
9.1. Normative References . . . . .	18
9.2. Informative References . . . . .	19
9.3. External Informative References . . . . .	20
Authors' Addresses . . . . .	20

## 1. Introduction

The nodes in a [IEEE802154e] TSCH network follow a communication schedule. The entity (centralized or decentralized) responsible for building and maintaining that schedule has very precise control over the trade-off between the network's latency, bandwidth, reliability and power consumption. During early interoperability testing and development, however, simplicity is often more important than efficiency. One goal of this document is to define the simplest set of rules for building a [IEEE802154e] TSCH-compliant network, at the necessary price of lesser efficiency. Yet, this minimal mode of operation can also be used during network bootstrap before any schedule is installed into the network so nodes can self-organize and the management and configuration information be distributed. In addition, as outlined in [I-D.phinney-roll-rpl-industrial-applicability], the minimal configuration can be used as a fallback mode of operation, ensuring connectivity of nodes in case that dynamic scheduling mechanisms fail or are not available. [IEEE802154e] provides a mechanism whereby the details of slotframe length, timeslot timing, and channel hopping pattern are communicated at synchronization to a node, also Enhanced Beacons can be used to periodically update nodes information. This document describes specific settings for these parameters. Nodes MUST broadcast properly formed Enhanced Beacons to announce these values, but during initial implementation and debugging it may be convenient to preconfigure these values.

## 2. Minimal Schedule Configuration

In order to form a network, a minimum schedule configuration is required so nodes can advertise the presence of the network, and allow other nodes to join.

### 2.1. Slotframe

The slotframe, as defined in [I-D.ietf-6tisch-terminology], is an abstraction of the link layer that defines a collection of time slots of equal length, and which repeats over time. In order to set up a minimal TSCH network, nodes need to be synchronized with the same slotframe configuration so they can exchange Enhanced Beacons (EBs)

and data packets. This document recommends the following slotframe configuration.

Minimal configuration

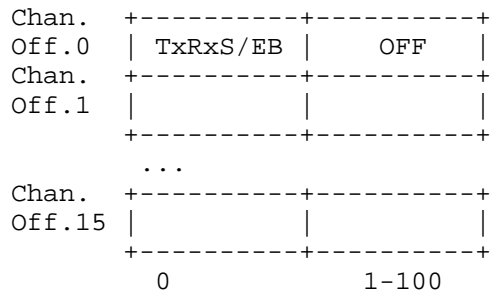
Property	Value
Number of time slots per Slotframe	Variable
Number of available frequencies	16
Number of scheduled cells	1 (slotOffset 0) (macLinkType NORMAL)
Number of unscheduled cells	The remainder of the slotframe
Number of MAC retransmissions (max)	3 (4 attempts to tx)

The slotframe is composed of a configurable number of time slots. Choosing the number of time slots per slotframe needs to take into account network requirements such as density, bandwidth per node, etc. In the minimal configuration, there is only a single active slot in slotframe, used to transmit data and EBs, and receive information. The trade-off between bandwidth, latency and energy consumption can be controlled by choosing a different slotframe length. The active slot MAY be scheduled at the slotoffset 0x00 and channeloffset 0x00 and MUST be announced in the EBs. EBs are sent using this active slot and are not acknowledged. Data packets, as described in Section 2.2 use the same active slot. Per [IEEE802154e], data packets sent unicast on this cell are acknowledged by the receiver. The remaining cells are unscheduled, and MAY be used by dynamic scheduling solutions. Details about such dynamic scheduling solution are out of scope.

The slotframe length (expressed in number of time slots) is configurable. The length used determines the duty cycle of the network. For example, a network with a 0.99% duty cycle is composed of a slotframe of 101 slots, which includes 1 active slot. The present document RECOMMENDS the use of a default slot duration set to 10ms and its corresponding default timeslot timings defined by the [IEEE802154e] macTimeslotTemplate. The use of the default macTimeslotTemplate MUST be announced in the EB by using the Timeslot IE containing only the default macTimeslotTemplateId. Other time slot durations MAY be supported and MUST be announced clearly. If

one uses a timeslot duration different than 10ms, it is RECOMMENDED to use a power-of-two of 10ms (i.e. 20ms, 40ms, 80ms, etc.). In this case, EBs MUST contain the complete TimeSlot IE as described in Section 2.4. This document also recommends to manufacturers to clearly indicate nodes not supporting the default timeslot value.

Example schedule with 0.99% duty cycle



EB: Enhanced Beacon

Tx: Transmit

Rx: Receive

S: Shared

OFF: Unscheduled (can be used by a dynamic scheduling mechanism)

## 2.2. Cell Options

Per the [IEEE802154e] TSCH, each scheduled cell has an associated bitmap of cell options, called LinkOption. The scheduled cell in the minimal schedule is configured as Hard cell [I-D.ietf-6tisch-tsch][I-D.ietf-6tisch-6top-interface]. Additional available cells can be scheduled by a dynamic scheduling solution. The dynamic scheduling solution is out of scope, and this specification does not make any restriction on the LinkOption associated with those dynamically scheduled cells (i.e. they can be hard cells or soft cells).

The active cell is assigned the bitmap of cell options below. Because both the "Transmit" and "Receive" bits are set, a node transmits if there is a packet in its queue, and listens otherwise. Because the "shared" bit is set, the back-off mechanism defined in [IEEE802154e] is used to resolve contention. This results in "Slotted Aloha" behavior. The "Timekeeping" flag is never set, since the time source neighbor is selected using the DODAG structure of the network (detailed below).

b0 = Transmit = 1 (set)

b1 = Receive = 1 (set)  
b2 = Shared = 1 (set)  
b3 = Timekeeping = 0 (clear)  
b4-b7 = Reserved (clear)

All remaining cells are unscheduled. In unscheduled cells, the nodes SHOULD keep their radio off. In a memory-efficient implementation, scheduled cells can be represented by a circular linked list. Unscheduled cells SHOULD NOT occupy any memory.

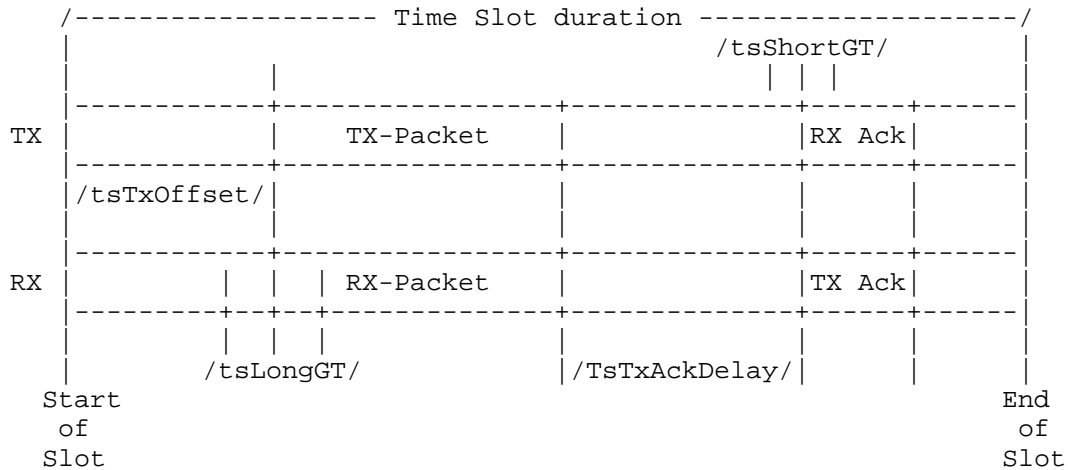
### 2.3. Retransmissions

The maximum number of link layer retransmissions is set to 3. For packets which require an acknowledgement, if none is received after a total of 4 attempts, the transmissions is considered failed and the link layer MUST notify the upper layer. Packets sent to the broadcast MAC address (including EBs) are not acknowledged and therefore not retransmitted.

### 2.4. Time Slot timing

The figure below shows an active timeslot in which a packet is sent from the transmitter node (TX) to the receiver node (RX). A link-layer acknowledgement is sent by the RX node to the TX node when the packet is to be acknowledged. The TsTxOffset duration defines the instant in the timeslot when the first byte of the transmitted packet leaves the radio of the TX node. The radio of the RX node is turned on TsLongGT/2 before that instant, and listens for at least TsLongGT. This allows for a de-synchronization between the two node of at most TsLongGT. The RX node needs to send the first byte of the MAC acknowledgement exactly TsTxAckDelay after the end of the last byte of the received packet. TX's radio has to be turned on TsShortGT/2 before that time, and keep listening for at least TsShortGT.

Time slot internal timing diagram



A 10ms time slot length is the default value defined by [IEEE802154e]. Section 6.4.3.3.3 of the [IEEE802154e] defines a default `macTimeslotTemplate`, i.e. the different duration within the slot. These values are summarized in the following table and MUST be used when utilizing the default time slot duration. In this case, the Timeslot IE only transports the `macTimeslotTemplateId` (0x00) as the timing values are well-known. If a timeslot template other than the default is used, the EB MUST contain a complete Timeslot IE indicating the timeslot duration and the corresponding timeslot timings, requiring 25 bytes.

Default timeslot durations (per [IEEE802154e], Section 6.4.3.3.3)

IEEE802.15.4e TSCH parameter	Value
TsTxOffset	2120us
TsLongGT	2000us
TsTxAckDelay	1000us
TsShortGT	400us
Time Slot duration	10000us

### 3. Enhanced Beacons Configuration and Content

[IEEE802154e] does not define how often EBs are sent, not their contents. The choice of the duration between two EBs needs to take into account whether EBs are used as the only mechanism to synchronize devices, or whether a Keep-Alive (KA) mechanism is also used. For a minimal TSCH configuration, a mote SHOULD send an EB every EB\_PERIOD. For additional reference see [I-D.ietf-6tisch-tsch] where different synchronization approaches are summarized.

EBs MUST be sent with the Beacon IEEE802.15.4 frame type and this EBs MUST carry the Information Elements (IEs) listed below.

The content of the IEs is presented here for completeness, however this information is redundant with [I-D.ietf-6tisch-tsch] and [IEEE802154e].

#### 3.1. Sync IE

Contains synchronization information such as ASN and Join Priority. The value of Join Priority is discussed in Section 5.2.

##### 3.1.1. IE Header

Length (b0-b7) = 0x06

Sub-ID (b8-b14) = 0x1a

Type (b15) = 0x00 (short)

##### 3.1.2. IE Content

ASN Byte 1 (b16-b23)

ASN Byte 2 (b24-b31)

ASN Byte 3 (b32-b39)

ASN Byte 4 (b40-b47)

ASN Byte 5 (b48-b55)

Join Priority (b56-b63)

### 3.2. TSCH Timeslot IE

Contains the timeslot template identifier. This specification uses the default timeslot template as defined in [IEEE802154e], Section 5.2.4.15.

#### 3.2.1. IE Header

Length (b0-b7) = 0x01

Sub-ID (b8-b14) = 0x1c

Type (b15) = 0x00 (short)

#### 3.2.2. IE Content

Timeslot Template ID (b0-b7) = 0x00

### 3.3. Channel Hopping IE

Contains the channel hopping template identifier. This specification uses the default channel hopping template, as defined in [IEEE802154e], Section 5.2.4.16.

#### 3.3.1. IE Header

Length (b0-b7) = 0x01

Sub-ID (b8-b14) = 0x1d

Type (b15) = 0x00 (short)

#### 3.3.2. IE Content

Channel Hopping Template ID (b0-b7) = 0x00

### 3.4. Frame and Link IE

Each node MUST indicate the schedule in each EB through a Frame and Link IE. This enables nodes which implement [IEEE802154e] to configure their schedule as they join the network.

#### 3.4.1. IE Header

Length (b0-b7) = variable

Sub-ID (b8-b14) = 0x1b

Type (b15) = 0x00 (short)

#### 3.4.2. IE Content

# Slotframes (b16-b23) = 0x01

Slotframe ID (b24-b31) = 0x01

Size Slotframe (b32-b47) = variable

# Links (b48-b55) = 0x01

For the active cell in the minimal schedule:

Channel Offset (2B) = 0x00

Slot Number (2B) = 0x00

LinkOption (1B) = as described in Section 2.2

#### 4. Acknowledgment

Link-layer acknowledgment frames are built according to [IEEE802154e]. Data frames and command frames sent to a unicast MAC destination address request an acknowledgment. The acknowledgment frame is of type ACK (0x10). Each acknowledgment contains the following IE:

##### 4.1. ACK/NACK Time Correction IE

The ACK/NACK time correction IE carries the measured de-synchronization between the sender and the receiver.

###### 4.1.1. IE Header

Length (b0-b7) = 0x02

Sub-ID (b8-b14) = 0x1e

Type (b15) = 0x00 (short)

###### 4.1.2. IE Content

Time Synchronization Information and ACK status (b16-b31)

The possible values for the Time Synchronization Information and ACK status are described in [IEEE802154e] and reproduced in the following table:



ACK status and Time Synchronization Information.

ACK Status	Value
ACK with positive time correction	0x0000 - 0x07ff
ACK with negative time correction	0x0800 - 0x0fff
NACK with positive time correction	0x8000 - 0x87ff
NACK with negative time correction	0x8800 - 0x8fff

## 5. Neighbor information

[IEEE802154e] does not define how and when each node in the network keeps information about its neighbors. This document recommends to keep the following information in the neighbor table:

### 5.1. Neighbor Table

The exact format of the neighbor table is implementation-specific, but it SHOULD contain the following information for each neighbor:

Neighbor statistics:

numTx: number of transmitted packets to that neighbor

numTxAck: number of transmitted packets that have been acknowledged by that neighbor

numRx: number of received packets from that neighbor

The EUI64 of the neighbor.

Timestamp when that neighbor was heard for the last time. This can be based on the ASN counter or any other time base. Can be used to trigger a keep-alive message.

RPL rank of that neighbor.

A flag indicating whether this neighbor is a time source neighbor.

Connectivity statistics (e.g., RSSI), which can be used to determine the quality of the link.

In addition to that information, each node has to be able to compute some RPL Objective Function (OF), taking into account the neighbor and connectivity statistics. An example RPL objective function is the OF Zero as described in [RFC6552] and Section 7.1.1.

## 5.2. Time Source Neighbor Selection

Each node MUST select at least one time source neighbor among the nodes in its RPL routing parent set. When a node joins a network, it has no routing information. To select its time source neighbor, it uses the Join Priority field in the EB, as described in Section 5.2.4.13 and Table 52b of [IEEE802154e]. The Sync IE contains the ASN and 1 Byte field named Join Priority. The Join Priority of any node is equivalent to the result of the function DAGRank(rank) as defined by [RFC6550] and Section 7.1.1. The Join Priority of the DAG root is zero, i.e., EBs sent from the DAG root are sent with Join Priority equal to 0. A lower value of the Join Priority indicates that the device is the preferred one to connect to. When a node joins the network, it MUST NOT send EBs before having acquired a RPL rank. This avoids routing loops and matches RPL topology with underlying mesh topology. As soon as a node acquires a RPL rank (see [RFC6550] and Section 7.1.1), it SHOULD send Enhanced Beacons including a Sync IE with Join Priority field set to DAGRank(rank), where rank is the node's rank. If a node receives EBs from different nodes with equal Join Priority, the time source neighbor selection should be assessed by other metrics that can help determine the better connectivity link. Time source neighbor hysteresis SHOULD be used, according to the rules defined in Section 7.2.3. If connectivity to the time source neighbor is lost, a new time source neighbor MUST be chosen among the neighbors in the RPL routing parent set.

The decision for a node to select one Time Source Neighbor when multiple EBs are received is open to implementers. For example a node MAY wait until one EB from NUM\_NEIGHBOURS\_TO\_WAIT neighbors have been received to select the best Time Source Neighbor. This condition MAY apply unless a second EB is not received after MAX\_EB\_DELAY seconds. This avoids initial hysteresis when selecting a first Time Source Neighbor.

Optionally, some form of hysteresis SHOULD be implemented to avoid frequent changes in time source neighbors.

## 6. Queues and Priorities

[IEEE802154e] does not define the use of queues to handle upper layer data (either application or control data from upper layers). This

document recommends the use of a single queue with the following rules:

When the node is not synchronized to the network, higher layers are not able to insert packets into the queue.

Frames generated by the MAC layer (e.g., EBs and ACK) have a higher priority than packets received from a higher layer.

IEEE802.15.4 frames of types Beacon and Command have a higher priority than IEEE802.15.4 frames of types Data and ACK.

One entry in the queue is reserved at all times for an IEEE802.15.4 frames of types Beacon or Command frames.

## 7. RPL on TSCH

Nodes in the network MUST use the RPL routing protocol [RFC6550].

### 7.1. RPL Objective Function Zero

Nodes in the network MUST use the RPL routing protocol [RFC6550] and implement the RPL Objective Function Zero [RFC6552].

#### 7.1.1. Rank computation

The rank computation is described at [RFC6552], Section 4.1. Briefly, a node rank is computed by the following equation:

$$R(N) = R(P) + \text{rank\_increase}$$

$$\text{rank\_increase} = (R_f * S_p + S_r) * \text{MinHopRankIncrease}$$

Where:

$R(N)$ : Rank of the node.

$R(P)$ : Rank of the parent obtained as part of the DIO information.

$\text{rank\_increase}$ : The result of a function that determines the rank increment.

$R_f$  ( $\text{rank\_factor}$ ): A configurable factor that is used to multiply the effect of the link properties in the  $\text{rank\_increase}$  computation. If none is configured,  $\text{rank\_factor}$  of 1 is used. In this specification, a  $\text{rank\_factor}$  of 1 MUST be used.

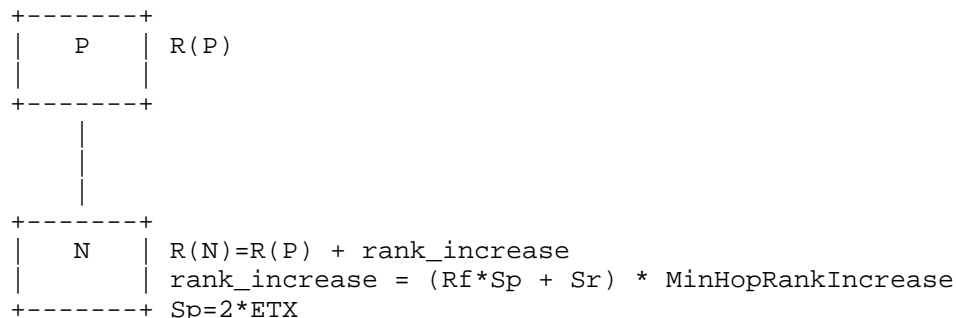
Sp (step\_of\_rank): (strictly positive integer) - an intermediate computation based on the link properties with a certain neighbor. In this specification, 2\*ETX (Expected Transmissions) as defined by [decouti03high] and [RFC6551] MUST be used. The ETX is computed as the inverse of the Packet Delivery Ratio (PDR), and MAY be computed as the number of acknowledged packets, divided by the number of transmitted packets to a certain node. E.g:  
 $Sp = 2 * numTX / numTXAck$

Sr (stretch\_of\_rank): (unsigned integer) - the maximum increment to the step\_of\_rank of a preferred parent, to allow the selection of an additional feasible successor. If none is configured to the device, then the step\_of\_rank is not stretched. In this specification, stretch\_of\_rank MUST be set to 0.

MinHopRankIncrease: the MinHopRankIncrease is set to the fixed constant DEFAULT\_MIN\_HOP\_RANK\_INCREASE [RFC6550].  
 DEFAULT\_MIN\_HOP\_RANK\_INCREASE has a value of 256.

DAGRank(rank): Equivalent to the floor of  $(Rf * Sp + Sr)$  as defined by [RFC6550]. Specifically, when an Objective Function computes Rank, this is defined as an unsigned integer (i.e., a 16-bit value) Rank quantity. When the Rank is compared, e.g. to determine parent relationships or loop detection, the integer portion of the Rank is used. The integer portion of the Rank is computed by the DAGRank() macro as  $\text{floor}(x)$  where  $\text{floor}(x)$  is the function that evaluates to the greatest integer less than or equal to  $x$ .  $\text{DAGRank}(\text{rank}) = \text{floor}(\text{rank} / \text{MinHopRankIncrease})$

#### Rank computation scenario

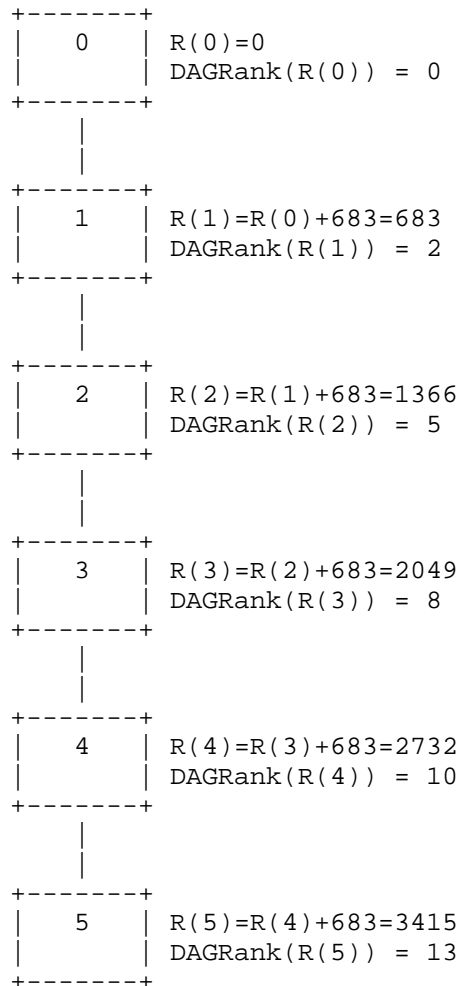


#### 7.1.2. Rank computation Example

This section illustrates with an example the use of the Objective Function Zero. Assume the following parameters:

$R_f = 1$  $S_p = 2 * ETX$  $S_r = 0$  $minHopRankIncrease = 256$  (default in RPL) $ETX = (numTX / numTXAck)$  $r(n) = r(p) + rank\_increase$  $rank\_increase = (R_f * S_p + S_r) * minHopRankIncrease$  $rank\_increase = 512 * numTx / numTxACK$

Rank computation example for 5 hop network where numTx=100 and numTxAck=75 for all nodes



## 7.2. RPL Configuration

In addition to the Objective Function (OF), a minimal configuration for RPL should indicate the preferred mode of operation and trickle timer operation so different RPL implementations can inter-operate. RPL information SHOULD be transported in the flow label in the LLN as defined in [I-D.thubert-6man-flow-label-for-rpl]

#### 7.2.1. Mode of Operation

For downstream route maintenance, in a minimal configuration, RPL SHOULD be set to operate in the Non-Storing mode as described by [RFC6550] Section 9.7. Storing mode ([RFC6550] Section 9.8) MAY be supported in less constrained devices.

#### 7.2.2. Trickle Timer

RPL signaling messages such as DIOs are sent using the Trickle Algorithm [RFC6550] (Section 8.3.1) and [RFC6206]. For this specification, the Trickle Timer MUST be used with the RPL defined default values [RFC6550] (Section 8.3.1). For a description of the Trickle timer operation see Section 4.2 on [RFC6206].

#### 7.2.3. Hysteresis

According to [RFC6552], [RFC6719] recommends the use of a boundary value (PARENT\_SWITCH\_THRESHOLD) to avoid constant changes of parent when ranks are compared. When evaluating a parent that belongs to a smaller path cost than current minimum path, the candidate node is selected as new parent only if the difference between the new path and the current path is greater than the defined PARENT\_SWITCH\_THRESHOLD. Otherwise the node MAY continue to use the current preferred parent. As for [RFC6719] the recommended value for PARENT\_SWITCH\_THRESHOLD is 192 when ETX metric is used, the recommendation for this document is to use PARENT\_SWITCH\_THRESHOLD equal to 394 as the metric being used is  $2 \cdot \text{ETX}$ . This is mechanism is suited to deal with parent hysteresis in both cases routing parent and time source neighbor selection.

#### 7.2.4. Variable Values

The following table presents the RECOMMENDED values for the RPL-related variables defined in the previous section.

## Recommended variable values

Variable	Value
EB_PERIOD	10s
MAX_EB_DELAY	180
NUM_NEIGHBOURS_TO_WAIT	2
PARENT_SWITCH_THRESHOLD	394

## 8. Acknowledgements

The authors would like to acknowledge the guidance and input provided by the 6TiSCH Chairs Pascal Thubert and Thomas Watteyne.

## 9. References

## 9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC6206] Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko, "The Trickle Algorithm", RFC 6206, March 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.
- [RFC6551] Vasseur, JP., Kim, M., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks", RFC 6551, March 2012.
- [RFC6552] Thubert, P., "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6552, March 2012.
- [RFC6719] Gnawali, O. and P. Levis, "The Minimum Rank with Hysteresis Objective Function", RFC 6719, September 2012.



## 9.2. Informative References

### [I-D.ietf-6tisch-tsch]

Watteyne, T., Palattella, M., and L. Grieco, "Using IEEE802.15.4e TSCH in an LLN context: Overview, Problem Statement and Goals", draft-ietf-6tisch-tsch-00 (work in progress), November 2013.

### [I-D.ietf-6tisch-architecture]

Thubert, P., Watteyne, T., and R. Assimiti, "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-architecture-02 (work in progress), June 2014.

### [I-D.ietf-6tisch-terminology]

Palattella, M., Thubert, P., Watteyne, T., and Q. Wang, "Terminology in IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-terminology-01 (work in progress), February 2014.

### [I-D.ietf-6tisch-6top-interface]

Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top) Interface", draft-ietf-6tisch-6top-interface-00 (work in progress), March 2014.

### [I-D.richardson-6tisch-security-architecture]

Richardson, M., "security architecture for 6top: requirements and structure", draft-richardson-6tisch-security-architecture-02 (work in progress), April 2014.

### [I-D.ietf-roll-terminology]

Vasseur, J., "Terms used in Routing for Low power And Lossy Networks", draft-ietf-roll-terminology-13 (work in progress), October 2013.

### [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.thubert-6man-flow-label-for-rpl]

Thubert, P., "The IPv6 Flow Label within a RPL domain", draft-thubert-6man-flow-label-for-rpl-03 (work in progress), May 2014.

## 9.3. 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.

[decouti03high]

De Couto, D., Aguayo, D., Bicket, J., and R. Morris, "A High-Throughput Path Metric for Multi-Hop Wireless Routing", MobiCom '03, The 9th ACM International Conference on Mobile Computing and Networking, San Diego, California", June 2003.

[OpenWSN]

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 , August 2012.

## Authors' Addresses

Xavier Vilajosana (editor)  
Universitat Oberta de Catalunya  
156 Rambla Poblenou  
Barcelona, Catalonia 08018  
Spain

Phone: +34 (646) 633 681  
Email: xvilajosana@uoc.edu

Kris Pister  
University of California Berkeley  
490 Cory Hall  
Berkeley, California 94720  
USA

Email: pister@eecs.berkeley.edu

6TiSCH  
Internet-Draft  
Intended status: Informational  
Expires: January 5, 2015

MR. Palattella, Ed.  
SnT/Univ. of Luxembourg  
P. Thubert  
cisco  
T. Watteyne  
Linear Technology / Dust Networks  
Q. Wang  
Univ. of Sci. and Tech. Beijing  
July 4, 2014

Terminology in IPv6 over the TSCH mode of IEEE 802.15.4e  
draft-ietf-6tisch-terminology-02

Abstract

6TiSCH proposes an architecture for an IPv6 multi-link 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. This document extends existing terminology documents available for Low-power and Lossy Networks to provide additional terminology elements.

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 5, 2015.

## Copyright Notice

Copyright (c) 2014 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. IANA Considerations . . . . .	9
4. Security Considerations . . . . .	9
5. Acknowledgments . . . . .	9
6. References . . . . .	9
6.1. Normative References . . . . .	9
6.2. Informative References . . . . .	10
6.3. External Informative References . . . . .	11
Authors' Addresses . . . . .	11

## 1. Introduction

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 in industrial automation or in-vehicle sensors and actuators.

At IEEE802.1, the "Audio/Video Task Group", was renamed TSN for Time Sensitive Networking. The IEEE802.15.4 Medium Access Control (MAC) has evolved with IEEE802.15.4e which provides in particular the Time Slotted Channel Hopping (TSCH) mode for industrial-type applications. Both provide deterministic capabilities to the point that a packet that pertains to a certain flow crosses the network from node to node following a very precise schedule, like a train leaves intermediate stations at precise times along its path.

This document provides additional terminology elements to cover terms that are new to the context of TSCH wireless networks and other deterministic networks.

## 2. Terminology

The draft extends [I-D.ietf-roll-terminology] and use terms from RFC 6550 [RFC6550] and RFC 6552 [RFC6552], which are all included here by reference.

The draft does not reuse terms from IEEE802.15.4e such as "path" or "link" which bear a meaning that is quite different from classical IETF parlance.

This document adds the following terms:

- 6TiSCH:** IPv6 over the Timeslotted Channel Hopping (TSCH) mode of IEEE 802.15.4e. It defines the 6top sublayer and a set of protocols (in particular, for setting up a schedule with a centralized or distributed approach, managing the resource allocation), as well as the architecture to bind them together, for use in IPv6 TSCH based networks.
- 6F:** IPv6 Forwarding. One of the three forwarding models supported by 6TiSCH. Packets are routed at layer 3, where Quality of Service (QoS) and Random Early Detection (RED) [RFC2309] operations are expected to prioritize flows with differentiated services.
- 6top:** 6top is the adaptation sublayer between TSCH and upper layers like 6LoWPAN and RPL. It is defined in [I-D.wang-6tisch-6top-sublayer].
- 6top Data Convey Model:** Model describing how the 6top adaptation layer feeds the data flow coming from upper layers into TSCH. It is composed by an I-MUX module, a MUX module, a set of priority queues, and a PDU (Payload Data Unit). See [I-D.wang-6tisch-6top-sublayer].
- ASN:** Absolute Slot Number, the total number of timeslots that has elapsed since the start of the network or an arbitrary start time (i.e., a timeslot counter, incremented by one at each timeslot). It is wide enough to not roll over in practice. See [IEEE802154e].
- Blacklist:** Set of frequencies which should not be used for communication.
- BBR:** Backbone Router. In the 6TiSCH architecture, it is an LBR and also a IPv6 ND-efficiency-aware Router (NEAR) [I-D.chakrabarti-nordmark-6man-efficient-nd]. It performs ND proxy operations between registered devices

and classical ND devices that are located over the backbone.

**Broadcast cell:** A scheduled cell used for broadcast transmission.

**Bundle:** A group of equivalent scheduled cells, i.e. cells identified by different [slotOffset, channelOffset], which are scheduled for a same purpose, with the same neighbor, with the same flags, and the same slotframe. The size of the bundle refers to the number of cells it contains. Given the length of the slotframe, the size of the bundle translates directly into bandwidth.

**Cell:** A single element in the TSCH schedule, identified by a slotOffset, a channelOffset, a slotframeHandle. A cell can be scheduled or unscheduled.

**ChannelOffset:** Identifies a row in the TSCH slotframe. The number of available channelOffsets is equal to the number of available frequencies. The channelOffset translates into a frequency when the communication takes place, resulting in channel hopping, as detailed in [I-D.ietf-6tisch-tschan].

**Channel distribution/usage (CDU) matrix:** : Matrix of height equal to the number of available channels (i.e, ChannelOffsets), representing the spectrum (channel) distribution among the different (RPL parent) nodes in the networks. Every single element of the matrix belongs to a specific chunk. It has to be noticed that such matrix, even though it includes all the cells grouped in chunks, belonging to different slotframes, is different from the TSCH schedule.

**Chunk:** A well-known list of cells, well-distributed in time and frequency, within a CDU matrix; a chunk represents a portion of a CDU matrix that is globally known by all the nodes in the network, with typically at most one cell per slotOffset for single radio devices. Once appropriated, a chunk can be managed separately by a single node within its interference domain. A node may appropriate multiple chunks, and use them according to a specific policy. Chunks may overlap. They can be pre-programmed, or can be computed by an external entity at the network bootstrap.

Chunk ownership appropriation: The process by which an individual node obtains a chunk to manage based on peer-to-peer interaction with its neighbors.

Chunk ownership delegation: The process by which an individual node obtains a chunk to manage based on point-to-point interaction with an external entity.

Communication Paradigm: It is Associated with the Information Model [RFC3444] of the state that is exchanged, and indicates: the location of that state (e.g., centralized vs. distributed, RESTful, etc.), the numbers of parties (e.g., P2P vs. P2MP) and the relationship between parties (e.g., master/slave vs. peers) at a high level of protocol abstraction. Layer 5 client/server REST is a typical communication paradigm, but industrial protocols also use publish/subscribe which is P2MP and source/sink which is MP2MP and primarily used for alarms and alerts at the application layer. At layer 3, basic flooding, P2P synchronization and path-marking (RSVP-like) are commonly used paradigms, whereas at layer 2, master/slave polling and peer-to-peer forwarding are classical examples.

Dedicated Cell: A cell that is reserved for a given node to transmit to a specific neighbor.

Distributed cell reservation: A reservation of a cell done by one or more in-network entities (typically a connection endpoint).

Distributed track reservation: A reservation of a track done by one or more in-network entities (typically a connection endpoint).

EB: Enhanced Beacon frame used by a node to announce the presence of the network. It contains information about the timeslot length, the current ASN value, the slotframes and timeslots the beaconing mote is listening on, and a 1-byte join priority (i.e., number of hops separating the node sending the EB, and the PAN coordinator).

FF: 6LoWPAN Fragment Forwarding. It is one of the three forwarding models supported by 6TiSCH. The 6LoWPAN Fragment is used as a label for switching at the 6LoWPAN sublayer, as defined in [I-D.thubert-roll-forwarding-frags].

- GMPLS:** Generalized Multi-Protocol Label Switching, a 2.5 layer service that is used to forward packets based on the concept of generalized labels.
- Hard Cell:** A scheduled cell which the 6top sublayer cannot reallocate. See [I-D.wang-6tisch-6top-sublayer].
- Hopping Sequence:** Ordered sequence of frequencies, identified by a `Hopping_Sequence_ID`, used for channel hopping, when translating the channel offset value into a frequency (i.e., PHY channel). See [IEEE802154e] and [I-D.ietf-6tisch-tsch].
- IE:** Information Elements, a list of Type-Length-Value containers placed at the end of the MAC header, used to pass data between layers or devices. A small number of types are defined by [IEEE802154e], but a range of types is available for extensions, and thus, is exploitable by 6TiSCH. See [IEEE802154e].
- I-MUX module:** Inverse-Multiplexer, a classifier that receives 6LoWPAN frames and places them into priority queues. See [I-D.wang-6tisch-6top-sublayer].
- Interaction Model:** It is a particular way of implementing a communication paradigm. Defined at a lower level of abstraction, it includes protocol-specific details such as a particular method (e.g., a REST GET) and a Data Model for the state to be exchanged.
- KMP:** Key Managment Protocol.
- LBR:** LLN Border Router. It is an LLN device, usually powered, that acts as a Border Router to the outside within the 6TiSCH architecture.
- Link:** A communication facility or medium over which nodes can communicate at the link layer, i.e., the layer immediately below IP. Thus, the IETF parlance for the term "Link" is adopted, as opposed to the IEEE802.15.4e terminology. In the context of the 6TiSCH architecture, which applies to Low Power Lossy Networks (LLNs), an IPv6 subnet is usually not congruent to a single link and techniques such as IPv6 Neighbor Discovery Proxying are used to achieve reachability within the multilink subnet. A link is distinct from a track. In fact, link local addresses are not expected to be used over a track for end to end communication. Finally, from the Layer 3



perspective (where the inner complexities of TSCH operations are hidden to enable classical IP routing and Forwarding), a single radio interface may be seen as a number of Links with different capabilities for unicast or multicast services.

**Logical Cell:** A cell that corresponds to granted bandwidth but is only lazily associated to a physical cell, based on usage.

**MAC:** Medium Access Control.

**MUX module:** Multiplexer, the entity that dequeues frames from priority queues and associates them to a cell for transmission. See [I-D.wang-6tisch-6top-sublayer].

**NEAR:** Energy Aware Default Router, as defined in [I-D.chakrabarti-nordmark-6man-efficient-nd].

**NME:** Network Management Entity, the entity in the network managing cells and other device resources. It may cooperate with the PCE. It interacts with LLN nodes through the backbone router.

**PANA:** Protocol for carrying Authentication for Network Access, as defined in [RFC5191] .

**PCE:** Path Computation Element, the entity in the network which is responsible for building and maintaining the TSCH schedule, when centralized scheduling is used.

**PCE cell reservation:** The reservation of a cell done by the PCE.

**PCE track reservation:** The reservation of a track done by the PCE.

**QoS:** Quality of Service.

**(to) reallocate a cell:** The action operated by the 6top sublayer of changing the slotOffset and/or channelOffset of a soft cell.

**SA:** Security Association.

**(to) Schedule a cell:** The action of turning an unscheduled cell into a scheduled cell.

**Scheduled cell:** A cell which is assigned a neighbor MAC address (broadcast address is also possible), and one or more of

the following flags: TX, RX, shared, timeskeeping. A scheduled cell can be used by the IEEE802.15.4e TSCH implementation to communicate. A scheduled cell can be a hard cell or a soft cell.

**Shared Cell:** A cell marked with both the "TX" and "shared" flags. This cell can be used by more than one transmitter node. A backoff algorithm is used to resolve contention. See [I-D.ietf-6tisch-tsch].

**SlotOffset:** Identifies a column in the TSCH schedule, i.e., the number of timeslots since the beginning of the current iteration of the slotframe.

**Slotframe:** A MAC-level abstraction that is internal to the node and contains a series of timeslots of equal length and priority. It is characterized by a slotframe\_ID, and a slotframe\_size. Multiple slotframes can coexist in a node's schedule, i.e., a node can have multiple activities scheduled in different slotframes, based on the priority of its packets/traffic flows. The timeslots in the Slotframe are indexed by the SlotOffset; the first timeslot is at SlotOffset 0.

**Soft Cell:** A scheduled cell which the 6top sublayer can reallocate, as described in [I-D.wang-6tisch-6top-sublayer].

**TF:** Track Forwarding. It is the simplest and fastest forwarding model supported by 6TiSCH. It is a G-MPLS-like forwarding model. The input cell characterizes the flow and indicates the output cell.

**Timeslot:** A basic communication unit in TSCH which allows a transmitter node to send a frame to a receiver neighbor, and that receiver neighbor to optionally send back an acknowledgment.

**Time Source Neighbor:** A neighbor a node uses as its time reference, and to which it needs to keep its clock synchronized. A node can have one or more time source neighbors.

**Track:** A determined sequence of cells along a multi-hop path. It is typically the result of a reservation. The node that initializes the process for establishing a track is the owner of the track. The latter assigns a unique identifier to the track, called TrackID.

TrackID: Unique identifier of a track, assigned by the owner of the track.

TSCH: Time Slotted Channel Hopping, a medium access mode of the [IEEE802154e] standard which uses time synchronization to achieve ultra low-power operation and channel hopping to enable high reliability.

TSCH Schedule: A matrix of cells, each cell indexed by a slotOffset and a channelOffset. The TSCH schedule contains all the scheduled cells from all slotframes and is sufficient to qualify the communication in the TSCH network. The "width of the matrix is equal to the number of scheduled timeslots in all the concurrent active slotframes. The number of channelOffset values (the "height" of the matrix) is equal to the number of available frequencies.

unscheduled cell: A cell which is not used by the IEEE802.15.4e TSCH implementation.

### 3. IANA Considerations

This specification does not require IANA action.

### 4. Security Considerations

This specification is not found to introduce new security threats.

### 5. Acknowledgments

Thanks to the IoT6 European Project (STREP) of the 7th Framework Program (Grant 288445).

### 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.

[RFC2309] Braden, B., Clark, D., Crowcroft, J., Davie, B., Deering, S., Estrin, D., Floyd, S., Jacobson, V., Minshall, G., Partridge, C., Peterson, L., Ramakrishnan, K., Shenker, S., Wroclawski, J., and L. Zhang, "Recommendations on Queue Management and Congestion Avoidance in the Internet", RFC 2309, April 1998.

- [RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", RFC 3444, January 2003.
- [RFC5191] Forsberg, D., Ohba, Y., Patil, B., Tschafenig, H., and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA)", RFC 5191, May 2008.
- [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.
- [RFC6552] Thubert, P., "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6552, March 2012.

## 6.2. Informative References

- [I-D.chakrabarti-nordmark-6man-efficient-nd]  
Chakrabarti, S., Nordmark, E., Thubert, P., and M. Wasserman, "IPv6 Neighbor Discovery Optimizations for Wired and Wireless Networks", draft-chakrabarti-nordmark-6man-efficient-nd-05 (work in progress), February 2014.
- [I-D.ietf-6tisch-6top-interface]  
Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top) Interface", draft-ietf-6tisch-6top-interface-00 (work in progress), March 2014.
- [I-D.ietf-6tisch-architecture]  
Thubert, P., Watteyne, T., and R. Assimiti, "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-architecture-02 (work in progress), June 2014.
- [I-D.ietf-6tisch-coap]  
Sudhaakar, R. and P. Zand, "6TiSCH Resource Management and Interaction using CoAP", draft-ietf-6tisch-coap-00 (work in progress), May 2014.
- [I-D.ietf-6tisch-minimal]  
Vilajosana, X. and K. Pister, "Minimal 6TiSCH Configuration", draft-ietf-6tisch-minimal-01 (work in progress), June 2014.

[I-D.ietf-6tisch-tsch]

Watteyne, T., Palattella, M., and L. Grieco, "Using IEEE802.15.4e TSCH in an LLN context: Overview, Problem Statement and Goals", draft-ietf-6tisch-tsch-00 (work in progress), November 2013.

[I-D.ietf-roll-terminology]

Vasseur, J., "Terms used in Routing for Low power And Lossy Networks", draft-ietf-roll-terminology-13 (work in progress), October 2013.

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

Thubert, P. and J. Hui, "LLN Fragment Forwarding and Recovery", draft-thubert-roll-forwarding-frags-02 (work in progress), September 2013.

[I-D.wang-6tisch-6top-sublayer]

Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top)", draft-wang-6tisch-6top-sublayer-00 (work in progress), February 2014.

### 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.

### Authors' Addresses

Maria Rita Palattella (editor)  
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

Pascal Thubert  
Cisco Systems, Inc  
Village d'Entreprises Green Side  
400, Avenue de Roumanille  
Batiment T3  
Biot - Sophia Antipolis 06410  
France

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

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

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

Qin Wang  
Univ. of Sci. and Tech. Beijing  
30 Xueyuan Road  
Beijing, Hebei 100083  
China

Phone: +86 (10) 6233 4781  
Email: wangqin@ies.ustb.edu.cn

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: January 5, 2015

M. Richardson  
SSW  
July 4, 2014

6tisch secure join using 6top  
draft-richardson-6tisch--security-6top-01

Abstract

This document details a security architecture that permits a new 6tisch compliant node to join an 802.15.4e network. The process bootstraps the new node authenticating the node to the network, and the network to the node, and configuring the new node with the required 6tisch schedule. Any resemblance to WirelessHART/IEC62591 is entirely intentional.

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 5, 2015.

Copyright Notice

Copyright (c) 2014 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 . . . . .	3
1.1. Assumptions . . . . .	3
2. Terminology and Roles . . . . .	4
3. Architectural requirements of join protocol . . . . .	5
3.1. prefixes to use for join traffic . . . . .	8
4. security requirements . . . . .	8
4.1. threat model . . . . .	8
4.1.1. threats to the joining node . . . . .	8
4.1.2. threats to the resources of the network . . . . .	9
4.1.3. threats to other joining nodes . . . . .	10
4.2. implementation cost . . . . .	10
4.3. denial of service . . . . .	10
5. protocol requirements/constraints/assumptions . . . . .	10
5.1. inline/offline . . . . .	10
6. time sequence diagram . . . . .	10
6.1. explanation of each step . . . . .	11
6.1.1. step (1): enhanced beacon . . . . .	12
6.1.2. step (1B): send router solicitation . . . . .	12
6.1.3. step (1C): receive router advertisement . . . . .	12
6.1.4. step (2): acquire authorizer key . . . . .	12
6.1.5. step (3): receive authorizer key . . . . .	12
6.1.6. step (4): join request . . . . .	13
6.1.7. step (5): NS duplicate address request (DAR) . . . . .	13
6.1.8. step (7): 6LBR informs JCE of new node . . . . .	13
6.1.9. step (8): JCE informs/acks to 6LBR of new node . . . . .	13
6.1.10. step (9): NS duplicate address confirmation (DAC) . . . . .	13
6.1.11. step (10): JCE initiates connection to joining node . . . . .	13
6.1.12. step (11): Join Assistant forwards packet to joining node . . . . .	13
6.1.13. step (12): Joining node replies . . . . .	13
6.1.14. step (13): Join Assistant forwards reply to JCE . . . . .	13
6.2. size of each packet . . . . .	14
7. resulting security properties obtained from this process . . . . .	14
8. deployment scenarios underlying protocol requirements . . . . .	14
9. device identification . . . . .	14
9.1. PCE/Proxy vs Node identification . . . . .	14



9.2. Time source authentication / time validation . . . . .	14
9.3. description of certificate contents . . . . .	14
9.4. privacy aspects . . . . .	14
10. slotframes to be used during join . . . . .	15
11. configuration aspects . . . . .	15
12. authorization aspects . . . . .	15
12.1. how to determine a proxy/PCE from a end node . . . . .	15
12.2. security considerations . . . . .	15
13. security architecture . . . . .	15
14. Posture Maintenance . . . . .	15
15. Security Considerations . . . . .	15
16. Other Related Protocols . . . . .	15
17. IANA Considerations . . . . .	15
18. Acknowledgements . . . . .	15
19. References . . . . .	15
19.1. Normative references . . . . .	15
19.2. Informative references . . . . .	17
Author's Address . . . . .	17

## 1. Introduction

A challenging part with constructing an LLN with nodes from multiple vendors is providing enough security context to each node such that the network communication can form and remain secure. Most LLNs are small and have no operator interfaces at all, and even if they have debug interfaces (such as JTAG) with personnel trained to use that, doing any kind of interaction involving electrical connections in a dirty environment such as a factory or refinery is hopeless.

It is necessary to have a way to introduce new nodes into a 6tisch network that does not involve any direct manipulation of the nodes themselves. This act has been called "zero-touch" provisioning, and it does not occur by chance, but requires coordination between the manufacturer of the node, the service operator running the LLN, and the installers actually taking the devices out of the shipping boxes.

### 1.1. Assumptions

For the process described in this document to work, some assumptions about available infrastructure are made. These are perhaps more than assumptions, but rather architectural requirements; the exact operation of said infrastructure to be defined in a subsequent document.

In the diagrams and text that follows entities are named (and defined in the terminology section). Unless otherwise stated these are roles, not actual machines/systems. The roles are separated by network protocols in order that they roles can be performed by

different systems, not because they have to be. Different deployments will have different scaling requirements for those entities. Smaller deployments might co-located many roles together into a single ruggedized platform, while other deployments might operate all of the roles on distinct, multiply-redundant server classes located in a fully equipped datacentre.

## 2. Terminology and Roles

Most terminology should be taken from [I-D.ietf-6tisch-architecture] and from [I-D.ietf-6tisch-6top-interface] and [I-D.wang-6tisch-6top-sublayer]. As well, many terms are taken from [RFC6775].

The following roles/things are defined:

PCE	the Path Computation Engine. This entity reaches out to each of the nodes in the LLN, and configures an appropriate schedule using 6top.
Authz Server/ACE	the Authorization Server. This offloads calculation of access control lists and other access control decisions for constrained nodes. See [I-D.seitz-ace-problem-description]
JCE	the Join Coordination Entity. This acronym is chosen to parallel the PCE.
802.1AR	a certificate created according the specification in [IEEE.802.1AR]
joining node	The newly unboxed constrained node that needs to join a network.
join assistant	A constrained node near the joining node that will act as it's first 6LR, and will relay traffic to/from the joining node.
join network	A 802.15.4e network whose encryption and authentication key is "JOIN6TISCH".
production network	A 802.15.4e network whose encryption/authentication keys are determined by some algorithm. There may have network-wide group keys, or per-link keys.

### 3. Architectural requirements of join protocol

This section works from the ultimate goal, and goes backwards to prerequisite actions. Section 6 presents the protocol from beginning to end order.

The ultimate goal of the join protocol is to provide a new node with enough locally significant security credentials that it is able to take part in the network directly. The credentials may vary by deployment. They can be:

- 1) a network-wide shared symmetric key
- 2) a locally significant (one-level only) 802.11AR type DevID certificate

Given one of the the above, there are a number of possible protocols that can be used to generate layer-2 sessions keys for the node, including:

- 1) Mesh Link Exchange [I-D.kelsey-intarea-mesh-link-establishment]
- 2) work in 802.15.9
- 3) Security Framework and Key Management Protocol Requirements for 6TiSCH [I-D.ohba-6tisch-security] (this document provides the phase 0 required)
- 4) Layer-2 security aspects for the IEEE 802.15.4e MAC [I-D.piro-6tisch-security-issues]

The intermediate goal of the join protocol is to enable a Join Coordination Entity (JCE) to reach out to the new node, and install the credentials detailed above. The JCE must authenticate itself to the joining node so that the joining node will know that it has joined the correct network, and the joining node must authenticate itself to the JCE so that the JCE will know that this node belongs in the network. This two way authentication occurs in the 6top/CoAP/DTLS session that is established between the JCE and the joining node.

[I-D.ietf-6tisch-6top-interface] presents a way to interface to a 6top MIB. [I-D.ietf-6tisch-coap] explains how to access that MIB using CoAP. That model is to be extended to include security attributes for the network. The JCE would therefore reach out to the joining node and simply provision appropriate security properties into the joining node, much like the PCE will provision schedules.

This 6top-based secure join protocol has defined a push model for security provisioning by the JCE. This has been done for three reasons:

- 1) 6tisch nodes already have to have a 6top CoAP server for schedule provising
- 2) this permits the JCE to manage how many nodes are trying to join at the same time, and limit how much bandwidth/energy is used for the join operation, and also for the JCE to prioritize the join order for nodes.
- 3) making the JCE initiate the DTLS connection significantly simplifies the certificate chains that must be exchanged as the most constrained side (the joining node) provides it's credentials first, and lets the much richer JCE figure out what kind of certificate chain will be required to authenticate the JCE. In EAP-TLS/802.1x situations, the TLS channel is created in the opposite direction, and it would have to complete in a tentative way, and then further authorization occur in-band.

In order for a 6top/DTLS/CoAP connection to occur between the JCE and the joining node, there needs to be end-to-end IPv6 connectivity between those two entities. The joining node will not participate in the route-over RPL mesh, but rather will be seen by the network as being a 6lowpan only leaf-node.

There are some alternatives to having full end to end connectivity which are discussed in the security considerations section.

The specific mechanism to enable end to end connectivity with the JCE are still open but will consist of one of:

- (1) IPIP tunnel between Join Assistant and JCE
- (2) using straight RPL routing: the Join Assistant sends a DAO
- (3) using a separate RPL DODAG for join traffic
- (4) establishing a specific multi-hop 6tisch track for join traffic for each Join Assistant

Of these mechanisms, the only one which does not require additional state on the Join Assistant (which is also a constrained device) is (1) and (2). Mechanism (2) additionally requires no specific state on the Join Assistant. Mechanism (2), in a non-storing DODAG requires additional state on the DODAG root (6LBR) only; while mechanism (1) requires a similar amount of state on the JCE. For

deployments where the JCE is part of the 6LBR, the amount of state is similar, but in any case, the 6LBR is assumed to be a non-constrained node.

As long as the Join Assistant does not do any kind of stateful firewalling, the IPIP tunnel and the DAO (2) method can be done by the Join Assistant statelessly. Upward traffic from the Join Network must be restricted to a 6tisch slotframe(s) to which join traffic is welcome, no tunnelling is necessary as the upwards routes are all in place. A destination address ACL on traffic from the Join Network restricts the Joining Nodes to sending traffic only to the address of the JCE. (If JCE and 6LBR are colocated, then this is the address in the ABRO, if they are not colocated, then this address needs to have been provisioning in the Join Assistant when it joined, or could be carried in a new RA option)

When using option (2), networks that have storing mode DODAGs will consume routing resources on all intermediate nodes between the Join Assistant and the DODAG root. This resource will be depleted without any authentication, and this threat is detailed below.

Continuing to work backwards, in order the JCE reach out to provision the Joining Node, it needs to know that the new node is present. This is done by taking advantage of the 6lowPAN Address Resolution Option (ARO) (section 4.1 [RFC6775]). The ARO causes the new address to also be sent up to the 6LBR for duplicate detection using the DAR/DAC mechanism. The 6LBR simply needs to tell the JCE about this using a protocol that needs to be defined, but could be either DAR or NS.

In addition to needing to know the joining devices address from the DAR/NS, the JCE also needs to know the joining node's IDevID. If the IDevID is less than 64 bits, then it is possible that it could be placed into the EUI-64 option of the ARO, or the OUI of the [I-D.thubert-6lowpan-backbone-router] EARO. The JCE needs to know the joining node's IDevID to know if this is device that it should even attempt to provision; and if so, it may need to retrieve an appropriate certificate chain (see [I-D.richardson-6tisch-idevid-cert]) from the Factory in order for the JCE to prove it is the legitimate owner of the joining node.

Prior to being able to announce itself in a NS, the joining node needs to find the Join Network. This is done by listening to an extended beacon which are broadcast in designated slotframes by Join Assistants. The Extended Beacon provides a way for the Joining Node to synchronize itself to the overall timeslot schedule and provides an Aloha period in which the Joining Node can send a Router Solicitation, and receive an appropriate Router Advertisement giving

the Joining Node a prefix and default route to which to send join traffic.

It may be possible to eliminate a message exchange if space for a Router Advertisement can be found as part of the Join Network Extended Beacon. This Enhanced Beacon would be distinct to the Join Network, and would be encrypted with the well-known Join Network key.

### 3.1. prefixes to use for join traffic

What prefix would the joining node use for communication? There are three options:

- (1) just use link-local addresses (requires all traffic be tunneled)
- (2) use a prefix specifically for join traffic (may be easier with a join-only DODAG)
- (3) use the same prefix as the rest of the traffic (may require more complex ACLs, and leaks information to attackers)

## 4. security requirements

### 4.1. threat model

There are three kinds of threats that a join process must deal with: threats to the joining node, threats to the resources of the network, and threats to other joining nodes.

#### 4.1.1. threats to the joining node

A node may be taken out of its box by a malicious entity and powered on. This could happen during shipping, while being stored in a warehouse. The device may be subject to physical theft, or the goal of the attacker may be to turn the device into a trojan horse of some kind. Physical protection of the device is out of scope for this document; this document will henceforth assume that the device is sealed in some tamper-evident way and this document deals with attacks over the network.

An attacker may attempt to convince the joining node that it is the legitimate Production Network; this is done by putting up a legitimate looking Join Network, and following the protocol as described in this document. The Joining Node can not know if it has the correct Production Network until steps 11-13, when it attempts to validate the ClientCertificate provided by the JCE.

When the joining node determines that this is the incorrect network, it must remember the PANID of the network that it has attempted to join, and then look for another network to try. It SHOULD have some limit as the number of times it will try before going back to sleep, or shutting down, and it SHOULD take care not to consume more than some specified percentage of any battery it might have.

Should a malicious production network be present at the same time/place as the legitimate production network, a the malicious agent could intercept and replay various packets from the proper join network, but ultimately this either results in a jamming-like denial of service, and/or the the ClientCertificate will not validate.

It is a legitimate situation for there to be multiple possible join networks, and the joining node may have to try each one before it finds the network that it the right one for it. The incorrect, but non-malicious networks will not attempt the 6top provisioning step, and SHOULD return a negative result in steps 8/9, refusing the node's NS. Those incorrect networks will be recognize that the node does not belong to them, because they will be able to see the Joining Node's IDevID in the ARO of step 4.

#### 4.1.2. threats to the resources of the network

The production network has two important resources that may be attacked by malicious Joining nodes: 1) energy/bandwidth, 2) memory for routing entries.

A malicious joining node could send many NS messages to the Join Assistant (from many made up addresses), which would send many NS/DAR messages to the 6LBR, and this would consume bandwidth, and therefore energy from the members of the mesh along the path to the 6LBR. This can be mitigated by limited the total bandwidth available for joining.

A malicious joining node could send many NS messages, and if the 6LBR agreed to accept the new node (by IDevID), then the Join Assistant would MAY inject routing information into mesh for the Joining node. Non-storing DODAGs store are routing information in the DODAG Root (probably the 6LBR), which is generally not a constrained node. Storing DODAGs store routing entries at all nodes up to the DODAG, and those are constrained nodes. Using a separate Join DODAG, and having that DODAG be non-storing will reduce any impact on intermediate nodes, but it does cause resources to be used for the second DODAG, and it may have a code impact if the nodes otherwise would not implement non-storing RPL.

#### 4.1.3. threats to other joining nodes

A joining node (or the nodes of a malicious network, co-located near the legitimate production network) may mount attacks on legitimate nodes which have not yet joined.

The malicious nodes may attempt to perform 6top operations against the joining node to keep it from being able to respond to the legitimate 6top session from the legitimate JCE. During the Join phase, the Joining node MUST have all other resources and protocols turned off, even if they would normally be accessible as read-only unauthenticated CoAP resources.

Malicious nodes could use the Join Network to mount various DTLS based attacks against the joining node, such as sending very long certificate chains to validate. One might think to limit the length of such chains, but as shown in [I-D.richardson-6tisch-idevid-cert] the chain may as long as the supplier chain, plus may include additional certificates due to resales of plants/equipment/etc. Validating from a trusted certificate down to the specific certificate which proves ownership would eliminate random certificate chains, but the attacker could just feed the joining node legitimate chains that it observed (and replayed) from the legitimate JCE. This does no good; the Joining node finds that the DTLS connection is invalid, but it may significantly run batteries down.

#### 4.2. implementation cost

(storage of security material, computational cost)

#### 4.3. denial of service

other communication impacts of security protocol mechanics

### 5. protocol requirements/constraints/assumptions

#### 5.1. inline/offline

dependencies on centralized or external functionality, inline and offline

#### 6. time sequence diagram



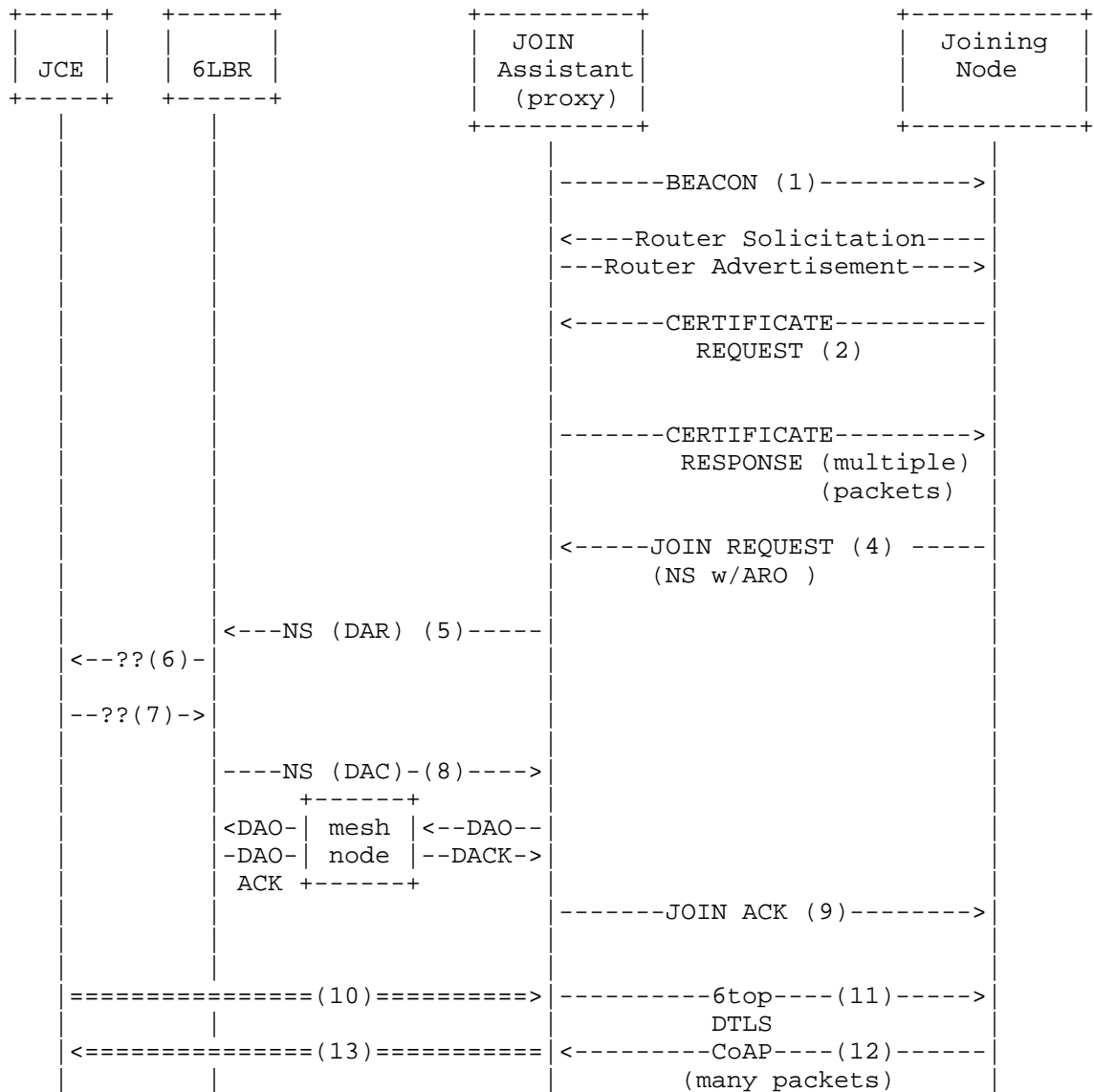


Figure 1: Message sequence for JOIN message

## 6.1. explanation of each step

#### 6.1.1. step (1): enhanced beacon

A 6tisch join/synchronization beacon is broadcast periodically, and is authenticated with a symmetric "beacon key":

- well known JOIN key, such "JOIN6TISCH"

- another key, provisioned in advance (OOB)

- a shared symmetric key derived from public part of top level certificate (a closely held "secret")

The purpose of this key is not to provide a high level of assurance, but rather to filter out 6tisch traffic from another random traffic that may be sharing the same radio frequencies.

These beacons are used for JOIN purpose only, and are not related to the Enhanced Beacons used in the rest of 6tisch.

#### 6.1.2. step (1B): send router solicitation

The joining node sends a router solicitation during the Aloha period of the beacon.

#### 6.1.3. step (1C): receive router advertisement

The joining node receives a router advertisement from the Join Assistant. It could include 6CO options to help compress packets, and should contain a prefix appropriate for join traffic.

#### 6.1.4. step (2): acquire authorizer key

Step (4) will involve doing a public key encryption to node performing the authorization management role. In order to do this, the new node needs to know the public key of the manager, and so in this step it requests that certificate from the neighbour that that it received the beacon from.

This step is optional, and it's benefit has not been demonstrated by a real world use case, but has been retained for now

#### 6.1.5. step (3): receive authorizer key

the proxy neighbour sends the key in one or more messages, along with the address of the authorizing server. The address of the authorization server could be an attribute of the certificate that is received.

## 6.1.6. step (4): join request

A regular Neighbour Solicitation is sent. This should contain an ARO (or EARO) option containing the Joining Nodes' IDevID. The ARO/EARO will be proxied by the Join Assistant as part of normal 6LowPAN processing for leaf nodes (non-RPL nodes) upwards to the 6LBR

## 6.1.7. step (5): NS duplicate address request (DAR)

## 6.1.8. step (7): 6LBR informs JCE of new node

## 6.1.9. step (8): JCE informs/acks to 6LBR of new node

The JCE could reply in the negative, and this would cause a DAC failure, TBD

## 6.1.10. step (9): NS duplicate address confirmation (DAC)

## 6.1.11. step (10): JCE initiates connection to joining node

The double lines indicate that an IPIP tunnel operation may be required. If a straight DAO or separate Join DODAG is used, then this is just a straight forwarding root to leaf node forwarding operation, and involves either using source routes (non-storing), or just forwarding for storing DODAGs.

A specific bandwidth allocation would be used for this join traffic

The production network encryption keys would be used for the join traffic

## 6.1.12. step (11): Join Assistant forwards packet to joining node

The JOIN Assistant would forward traffic to the Joining Node. Recognizing that this traffic the JOIN Network, the JOIN Assistant would use the JOIN Network key.

## 6.1.13. step (12): Joining node replies

The joining node replies, using JOIN Network key.

## 6.1.14. step (13): Join Assistant forwards reply to JCE

The JOIN Assistant, recognizing that the traffic came from the JOIN Network, restricts the destination that can be reached to the the JCE only. It can do this in a stateless way, and it does NOT need to track the traffic at (10) to open pinhole, etc.

Recognizing that the traffic came from the JOIN Network, the traffic would be placed into a bandwidth allocation (track?) that allows such traffic.

6.2. size of each packet

and number of frames needed to contain it.

7. resulting security properties obtained from this process

An end to end IPv6 CoAP/DTLS connection is created between the JCE and the Joining Node. This connection carries 6top commands to update security parameters. This results in either deployment of a single-level, locally relevant certificate, or deployment of a network-wide symmetric "Master Key"

8. deployment scenarios underlying protocol requirements

9. device identification

The JCE authenticates the joining node using a certificate chain provided inline during the DTLS negotiation. The certificate chain is rooted in a vendor certificate that the JCE must have preloaded, and is a statement as to the node's 802.1AR IDevID. The joining node authenticates the

9.1. PCE/Proxy vs Node identification

9.2. Time source authentication / time validation

Note: RPL Root authentication is a chartered item

9.3. description of certificate contents

9.4. privacy aspects

The EUI-64 of the Joining node is transmitted using a Well Known layer-2 encryption key. Within the ARO/EARO of the Neighbour Solicitation is an OUI, which may be identical to the EUI-64 of the Joining node, or it might be an unrelated IDevID.

An eavesdropper can therefore learn something about the manufacturer of every device as it joins.

10. slotframes to be used during join

how is this communicated in the (extended) beacon.

11. configuration aspects

(allocation of slotframes after join, network statistics, neighboetc.)

12. authorization aspects

lifecycle (key management, trust management)

12.1. how to determine a proxy/PCE from a end node

12.2. security considerations

what prevents a node from transmitting when it is not their turn  
(part one: jamming)

can a node successfully communicate with a peer at a time when not  
supposed to, may be tied to link layer security, or will it be  
policed by receiver?

13. security architecture

security architecture and fit of e.g. join protocol and provisioning  
into this

14. Posture Maintenance

(SACM related work)

15. Security Considerations

16. Other Related Protocols

17. IANA Considerations

18. Acknowledgements

19. References

19.1. Normative references

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

- [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.
- [IEEE.802.1AR]  
Institute of Electrical and Electronics Engineers, "Secure Device Identity", IEEE 802.1AR, 2009,  
<<http://www.ieee802.org/1/pages/802.1ar.html>>.
- [I-D.ietf-6tisch-coap]  
Sudhaakar, R. and P. Zand, "6TiSCH Resource Management and Interaction using CoAP", draft-ietf-6tisch-coap-00 (work in progress), May 2014.
- [I-D.ietf-6tisch-6top-interface]  
Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top) Interface", draft-ietf-6tisch-6top-interface-00 (work in progress), March 2014.
- [I-D.ietf-6tisch-architecture]  
Thubert, P., Watteyne, T., and R. Assimiti, "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-architecture-01 (work in progress), February 2014.
- [I-D.seitz-ace-problem-description]  
Seitz, L. and G. Selander, "Problem Description for Authorization in Constrained Environments", draft-seitz-ace-problem-description-00 (work in progress), May 2014.
- [I-D.richardson-6tisch-idevid-cert]  
Richardson, M., "X509.v3 certificate extension for authorization of device ownership", draft-richardson-6tisch-idevid-cert-00 (work in progress), May 2014.
- [I-D.wang-6tisch-6top-sublayer]  
Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top)", draft-wang-6tisch-6top-sublayer-00 (work in progress), February 2014.

## 19.2. Informative references

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

Thubert, P., "LoWPAN Backbone Router", draft-thubert-6lowpan-backbone-router-00 (work in progress), March 2008.

[I-D.kelsey-intarea-mesh-link-establishment]

Kelsey, R., "Mesh Link Establishment", draft-kelsey-intarea-mesh-link-establishment-05 (work in progress), February 2013.

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

Chasko, S., Das, S., Lopez, R., Ohba, Y., Thubert, P., and A. Yegin, "Security Framework and Key Management Protocol Requirements for 6TiSCH", draft-ohba-6tisch-security-01 (work in progress), March 2014.

[I-D.piro-6tisch-security-issues]

Piro, G., Boggia, G., and L. Grieco, "Layer-2 security aspects for the IEEE 802.15.4e MAC", draft-piro-6tisch-security-issues-02 (work in progress), June 2014.

## Author's Address

Michael C. Richardson  
Sandelman Software Works  
470 Dawson Avenue  
Ottawa, ON K1Z 5V7  
CA

Email: [mcr+ietf@sandelman.ca](mailto:mcr+ietf@sandelman.ca)  
URI: <http://www.sandelman.ca/>

6TiSCH  
Internet-Draft  
Intended status: Informational  
Expires: January 5, 2015

R. Struik  
Struik Security Consultancy  
July 4, 2014

6TiSCH Security Architectural Elements  
draft-struik-6tisch-security-architecture-elements-00

Abstract

This document describes security architectural elements that are relevant for the design of the 6TiSCH security architecture. (Note: this document is a work-in-progress and will provide more fine-tuned information with updated versions.)

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 5, 2015.

Copyright Notice

Copyright (c) 2014 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. Preliminaries . . . . .	2
1.1. Device Roles . . . . .	2
1.2. Initiator and Responder Model . . . . .	3
1.3. Cautionary Note - on Limitations of Cryptography . . . . .	3
1.4. Desired Protocol Properties . . . . .	3
1.5. Device Enrolment Phases . . . . .	4
1.6. Security Definitions . . . . .	5
1.7. Deployment Scenarios . . . . .	6
2. Security Considerations . . . . .	7
3. Other Related Protocols . . . . .	7
4. IANA Considerations . . . . .	7
5. Acknowledgements . . . . .	7
6. References . . . . .	7
6.1. Normative references . . . . .	7
6.2. Informative references . . . . .	8
Author's Address . . . . .	9

## 1. Preliminaries

### 1.1. Device Roles

When discussing security operations, it is useful to distinguish various device roles. Here, one should note that a device may assume more than one device role at the same time and that a particular role may be assumed by more than one device. Moreover, the mapping of device roles to devices may change over time (along a device's or network's lifecycle).

We distinguish the following roles:

1. Client. This device may move in and out of networks (that may be alien to it) and may have little network management functionality on board. Key words: nomadic, promiscuous, constrained.
2. Access point. This device may be more tied into a relatively stable infrastructure and may have more support for network management functionality or have reliable access hereto (e.g., via a back-end system). Key words: anchor, semi-stable connectivity, access portal.

3. Server. This device provides stable infrastructure and network management support, either intra-domain or inter domain (thereby, offering homogeneous or even heterogeneous functionality). Key words: core function, high availability, human-operator support.
4. CA. This device vouches for trust credentials, usually in offline way. Key words: trust anchor.

#### 1.2. Initiator and Responder Model

All peer-to-peer protocols are role-symmetrical (i.e., the role of initiator/responder roles are interchangeable). Protocols involving a third party assume communications with this third party to take place via the access point (since being the device more tied into infrastructure).

#### 1.3. Cautionary Note - on Limitations of Cryptography

Cryptographic techniques may provide logical assurances as to a device's identity, where and when communications originated, whom it was intended for, whom this can be read by, etc.

Cryptographic techniques do, however, only provide mechanical assurances and can generally not substitute human authorization decision elements (unless the latter are not important, such as with random, ad-hoc networks).

#### 1.4. Desired Protocol Properties

##### Security-Related:

1. Parties executing a security protocol should be explicitly aware of its security properties
2. Compromise of keys or devices should have limited effect on security of other devices or services
3. Attacks should not have a serious impact beyond the time interval/space during/in which these take place
4. Security protocols should minimize the impact of network outages, denial of service attacks

##### Communication Flows:

1. Security protocols should allow to be run locally, without third party involvement, if at all possible

2. The number of message exchanges for a joining client device should be reduced
3. Message exchanges should be structured so as to allow parallel execution of protocol steps, if possible

Computational Cost:

1. Security protocols should not impose an undue computational burden, esp. on joining client devices (An exception here may arise, when recovering from an event seriously impacting availability of the network.)

Device Capabilities:

1. Dependency on an accurate time-keeping mechanism should be reduced
2. Computational/time latency trade-offs should be tweaked to benefit those of joining client, if possible
3. Dependency on "homogeneous trust models" should be reduced, without jeopardizing security properties
4. Dependency on on-board trusted platforms and trusted I/O interfaces should be reduced

1.5. Device Enrolment Phases

1. Device Authentication. Client A and Access Point B authenticate each other and establish a shared key (so as to ensure on-going authenticated communications). This may involve server KDC as third party.
2. Authorization. Access Point B decides on whether/how to authorize device A (if denied, this may result in loss of bandwidth). Authorization decision may be delegated to server KDC or other 3rd-party device.
3. Configuration/Parameterization. Access Point B distributes configuration information to Client A, such as
  - \* IP address assignment info;
  - \* Bandwidth/usage constraints;
  - \* Scheduling info (including on re-authentication policy details)

This may originate from other network devices, for which it acts as proxy. This step may also include distribution of information from Client A to Access Point B and, more generally, synchronization of information between these two entities.

The device enrollment process is depicted in Figure Figure 1, where it is assumed that devices have access to certificates and where entities have access to the root CA key of its communicating parties (initial set-up requirement). Under these assumptions, the authentication step of the device enrollment process does not require online involvement of a third party.

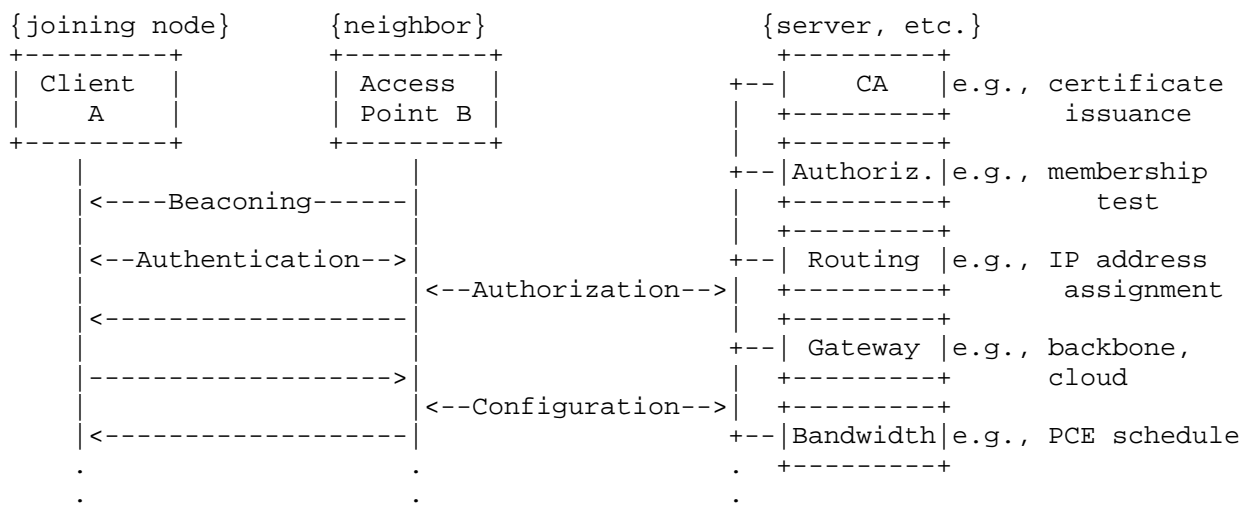


Figure 1: Networking Joining, with Only Authorization by Third Party

Aggressive scheme: Initiate authorization/configuration processes as soon as (presumed) device identity becomes available (invisible to Client A). Access Point B can deny bandwidth if authorization negative.

Note: Communication of configuration info depends on secure channel with Client A.

## 1.6. Security Definitions

1. Key Establishment: Protocol whereby a shared secret becomes available to two or more parties for subsequent cryptographic use

2. Key Transport: Key establishment technique where one party creates/obtains the secret and securely transfers it to other(s)
3. Key Agreement: Key establishment technique where the shared secret is derived based on information contributed by each of the parties involved, ideally so that no party can predetermine this secret value
4. Implicit Key Authentication: Assurance as to which specifically identified parties possibly may gain access to a specific key
5. Key Confirmation: Assurance that second (possibly unknown) party has possession of a particular key
6. Explicit Key Authentication: Combination of implicit key authentication and key confirmation
7. Unilateral Key Control: Key establishment protocol whereby one party can influence the shared secret
8. Forward Secrecy: Assurance that compromise of long-term keys does not compromise past session keys
9. Entity Authentication: Assurance of active involvement of second explicitly identified party in protocol
10. Mutual vs. Unilateral: Adjective indicating symmetry, resp. asymmetry, of assurances amongst parties
11. Identity Protection: Assurance as to which specifically identified parties may gain access to identity info
12. Certificate ? Credential that vouches for authenticity of binding between a public key and other information, including the identity of the owner of the public key in question
13. Key Possession? Assurance that a specific (possibly unknown) party has possession of a particular key

Esoteric properties: Unknown Key Share Resilience, Session Key Retrieval, Key Compromise Impersonation

#### 1.7. Deployment Scenarios

Deployment scenarios discussed with industrial control user community:

1. Scenario #1: mix-and-match of nodes from different vendors

2. Scenario #2: addition of nodes to operational network
3. Scenario #3: security audit
4. Scenario #4: device repair and replacement (roaming in/out different user sites)
5. Scenario #5: network separation (devices joining wrong network)
6. Scenario #6: thwarting malicious attacks by (former) insiders
7. Scenario #7: thwarting attacks by outsiders via insiders (held at 'gunpoint')
8. Scenario #8: addition of subsystem ('skid') assembled elsewhere to operational network

## 2. Security Considerations

This document is all about security.

## 3. Other Related Protocols

## 4. IANA Considerations

## 5. Acknowledgements

Discussions amongst participants in the 6TiSCH security conference calls to-date helped to shape this document.

## 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.
- [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.

- [RFC7250] Wouters, P., Tschofenig, H., Gilmore, J., Weiler, S., and T. Kivinen, "Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", RFC 7250, June 2014.
- [RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, June 2014.
- [I-D.ietf-6tisch-coap]  
Sudhaakar, R. and P. Zand, "6TiSCH Resource Management and Interaction using CoAP", draft-ietf-6tisch-coap-00 (work in progress), May 2014.
- [I-D.ietf-6tisch-architecture]  
Thubert, P., Watteyne, T., and R. Assimiti, "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-architecture-02 (work in progress), June 2014.
- [I-D.wang-6tisch-6top-sublayer]  
Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top)", draft-wang-6tisch-6top-sublayer-00 (work in progress), February 2014.
- [I-D.ietf-6tisch-6top-interface]  
Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top) Interface", draft-ietf-6tisch-6top-interface-00 (work in progress), March 2014.

## 6.2. Informative references

- [I-D.garcia-core-security]  
Garcia-Morchon, O., Kumar, S., Keoh, S., Hummen, R., and R. Struik, "Security Considerations in the IP-based Internet of Things", draft-garcia-core-security-06 (work in progress), September 2013.
- [I-D.ietf-dice-profile]  
Hartke, K. and H. Tschofenig, "A DTLS 1.2 Profile for the Internet of Things", draft-ietf-dice-profile-01 (work in progress), May 2014.
- [I-D.kumar-dice-dtls-relay]  
Kumar, S., Keoh, S., and O. Garcia-Morchon, "DTLS Relay for Constrained Environments", draft-kumar-dice-dtls-relay-01 (work in progress), April 2014.

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

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

[IEEE802.15.4-2011]

Institute for Electrical and Electronics Engineers, "IEEE 802.15.4-2011, IEEE Standard for Local and Metropolitan Area Networks - Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)", September 2011.

[IEEE802.15.4e-2012]

Institute for Electrical and Electronics Engineers, "IEEE 802.15.4e-2012, IEEE Standard for Local and Metropolitan Area Networks - Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs), Amendment 1: MAC Sublayer", April 2012.

[Wireless-HART]

International Electrotechnical Commission, "IEC 62591, Ed. 2.0: Industrial Communication Networks - Wireless Communication Network and Communication Profiles - WirelessHART (Draft)", November 2013.

[ISA100.11a]

International Electrotechnical Commission, "IEC 62734, Ed. 1: Industrial Communication Networks - Wireless Communication Network and Communication Profiles - ISA 100.11a (Draft)", May 2013.

[ZigBee-IP]

ZigBee Alliance, "ZigBee IP Specification (ZigBee Public Document 13-002r00)", February 2013.

#### Author's Address

Rene Struik  
Struik Security Consultancy  
  
Email: rstruik.ext@gmail.com



6TiSCH  
Internet-Draft  
Intended status: Informational  
Expires: January 5, 2015

Q. Wang, Ed.  
Univ. of Sci. and Tech. Beijing  
X. Vilajosana  
Universitat Oberta de Catalunya  
T. Watteyne  
Linear Technology  
R. Sudhaakar  
Cisco Systems  
P. Zand  
University of Twente  
July 4, 2014

Transporting CoAP Messages over IEEE802.15.4e Information Elements  
draft-wang-6tisch-6top-coapie-00

## Abstract

This document describes the format of "CoAP IE", an IEEE802.15.4e Information Element which allows CoAP messages to be transported as part of the IEEE802.15.4e header. This enables 6top-to-6top communication between neighbor nodes in a 6TiSCH network.

## 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 5, 2015.

## Copyright Notice

Copyright (c) 2014 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
1.1. Requirements Notation . . . . .	2
1.2. Context within 6TiSCH . . . . .	2
1.3. Motivation . . . . .	3
1.4. Status of this Document . . . . .	3
2. CoAP IE Format . . . . .	4
3. Softcell Negotiation Interface RPC Definition . . . . .	5
4. CoAP support . . . . .	7
4.1. URI setting . . . . .	7
4.2. CoAP Block option . . . . .	7
4.3. Management Interface Protocol . . . . .	8
4.4. Negotiation interface protocol . . . . .	8
4.5. Acknowledgement . . . . .	9
4.6. Observe . . . . .	9
5. Implementation Considerations . . . . .	10
6. References . . . . .	10
6.1. Normative References . . . . .	10
6.2. Informative References . . . . .	11
6.3. External Informative References . . . . .	12
Authors' Addresses . . . . .	12

## 1. Introduction

### 1.1. Requirements Notation

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].

### 1.2. Context within 6TiSCH

This document fits in the work done at the IETF 6TiSCH WG as follows:

- o [I-D.wang-6tisch-6top-sublayer] defines the operation of the 6top sublayer, which monitors and manages the communication schedule used in the [IEEE802154e] TSCH network.

- o [I-D.ietf-6tisch-6top-interface] defines the interface of the 6top sublayer using the YANG data modeling language [RFC6020].
- o [I-D.ietf-6tisch-coap] translates this YANG model in CoAP resources and interactions, allowing an Internet host (possibly but not necessarily constrained) to monitor and manage the 6top sublayer of a 6TiSCH device.
- o This document defines a method for transporting those CoAP messages as part of the IEEE802.15.4e header. It does so by defining a new IEEE802.15.4e Information Element called "CoAP IE". This allows a 6TiSCH node to monitor and manage the 6top sublayer and enables pairwise communication for signaling and control between neighbor nodes.

### 1.3. Motivation

The 6TiSCH architecture [I-D.ietf-6tisch-architecture] allows for both centralized and distributed monitoring and management of a 6TiSCH schedule. [I-D.ietf-6tisch-coap] defines the mechanisms necessary for the centralized case. The present document defines a mechanism enabling the communication of nodes in a 1 hop neighborhood, enabling a distributed approach.

In particular, it allows a node to monitor and manage its neighbor node's MIB. Through the CoAP IE defined in this document, a node sends link-layer frames to its neighbor which contain, as part of the link-layer header, the CoAP messages defined in [I-D.ietf-6tisch-coap]. This allows a node to interact with the 6top interface of its neighbor, in a way equivalent to an Internet host interacting with a 6TiSCH device over CoAP.

In addition, this document describe the frame formats and interaction between a node and its neighbor during softcell negotiation [I-D.wang-6tisch-6top-sublayer], through the addition of an Remote Procedure Call "RPC" element to the YANG model defined in [I-D.ietf-6tisch-6top-interface].

We call "6top-to-6top" communication the interaction between a node and its neighbor using the CoAP IE.

### 1.4. Status of this Document

The authors decided to present the CoAP IE as a separate document to request discussion and suggestions for improvement from the Internet community.

If the document gets support, and after suggestions for improvement have been integrated, the author propose to merge it in existing 6TiSCH I-Ds as follows:

- o Section 3 would go into [I-D.ietf-6tisch-6top-interface];
- o Section 4 would go into [I-D.ietf-6tisch-coap];
- o Section 2 and Section 5 would go into [I-D.wang-6tisch-6top-sublayer].

## 2. CoAP IE Format

The CoAP IE is a container for transporting CoAP messages as part of the IEEE802.15.4e header, as an Information Element. It is used by both the management interface and the softcell negotiation interface for 6top-to-6top communication.

This IE is not present in [IEEE802154e]; it is defined in this document.

Format of a CoAP IE.

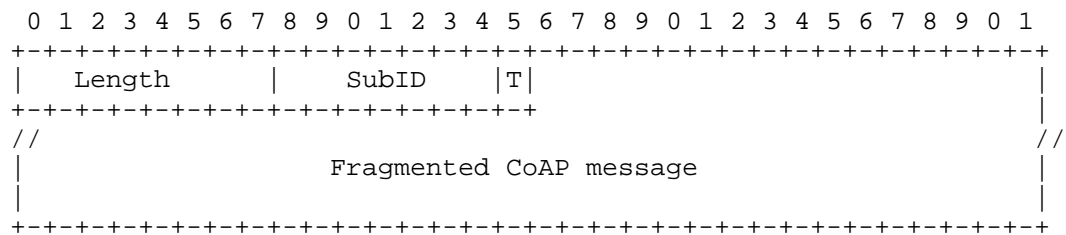


Figure 1

The fields in CoAP IE header are defined as follows.

- o Length = 1
- o SubID = 0x44
- o T = 0 (short type)

The content of CoAP IE is a CoAP message compliant to [RFC7252]. The CoAP message MAY use the CoAP Block option (see Section 4.2) in order to fragment large CoAP messages.

Format of CoAP IE with CoAP message.

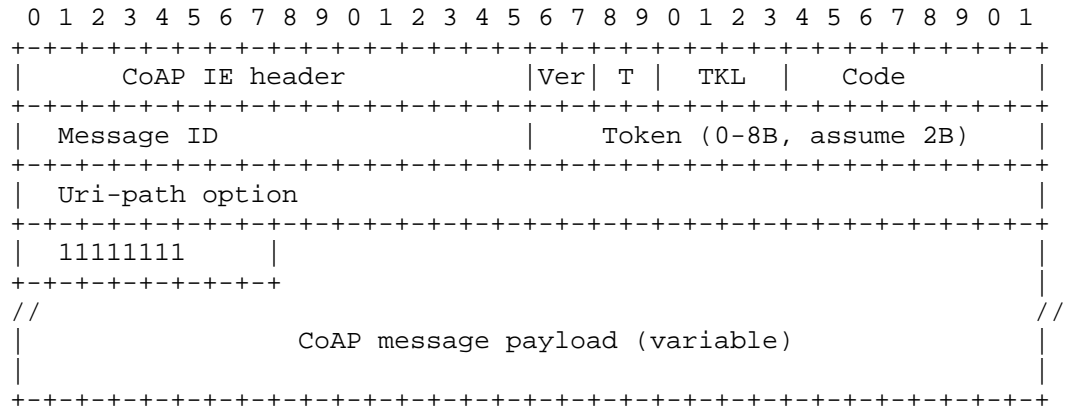


Figure 2

The Token Length (TKL) is set to 2;

Per [RFC7252], the Uri-path field consists of the following sub-fields:

- o Option Delta: 4bits, set to 11
- o Option Length: 4bits, set to 3
- o Option value: 3 bytes

The first byte of the option value is set to "6" (for 6top), "4" (for IEEE802.15.4), or "e" (for extension). The second and third bytes refer to the resource name in the corresponding group.

### 3. Softcell Negotiation Interface RPC Definition

This document proposes to replace the "6top Communication Protocol" defined in [I-D.wang-6tisch-6top-sublayer] by an extension to the YANG data model defined in [I-D.ietf-6tisch-6top-interface]. This allows neighbor nodes to negotiate the allocation of soft cells using the CoAP IE.

```
rpc softcell-negotiation {
  input {
    leaf Opcode {
      type enumeration {
        enum RESERVATION;
        enum REMOVE;
      }
    }
    leaf RequiredBW {
      type uint8;
    }
    leaf SlotframeID {
      type uint8;
    }
    leaf TrackID {
      type uint16;
      description
        "TrackID points to a tuple(TrackOwnerAddr,
        InstanceID)";
    }
    leaf NumofCandidate {
      type uint8;
    }
    List CandidateList {
      key "SlotOffset ChannelOffset";
      leaf SlotOffset{
        type uint16;
      }
      leaf ChannelOffset{
        type uint16;
      }
    }
  }
  output {
    leaf NumOfCells {
      type uint8;
    }
    List ResultedCells {
      key "SlotOffset ChannelOffset";
      leaf SlotOffset{
        type uint16;
      }
      leaf ChannelOffset{
        type uint16;
      }
    }
  }
}
```

## 4. CoAP support

### 4.1. URI setting

Uri-Host option = target node address;

Uri-Path option = 6t/6/[6top resource name], or 6t/4/[15.4 resource name], or 6t/e/[extension resource name], where [6top resource name] refers to the data resources or RPC defined by 6top, [15.4 resource name] refers to the data resources defined by IEEE802.15.4, and [extension resource name] refers to the data resources defined by an extensions of 6top, e.g. OTF. [6top resource name] , [15.4 resource name] and [extension resource name] are RECOMMENDED to be at most 2 bytes long.

### 4.2. CoAP Block option

In [I-D.ietf-core-block], two block options (Block1 and Block2) are defined to support block-wise transfers. The format of a fragmented message in a CoAP IE is defined as follows.

Format of CoAP IE content with fragmented message.

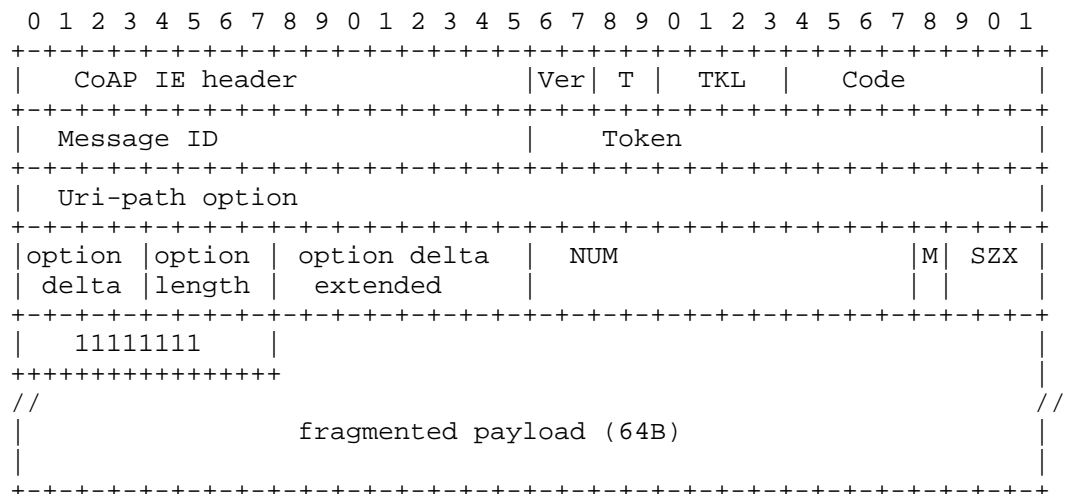


Figure 3

Per [I-D.ietf-core-block], the option Delta is 23 for Block1 and 27 for Block2. Related sub-fields are defined as follows.

- o Option delta: 4bits, set to 13, indicates an 8-bit unsigned integer follows the initial byte and the Option Delta minus 13.

- o Option length: 4bits, set to 2.
- o Option delta extended: 8bits, 23-13=10 and 27-13=14 for Block1 and Block2, respectively.

Per [IEEE802154], assuming the IE size constraint is 81 bytes, the related fields of the block option are defined as follows.

- o The size of the block (SZX): 3 bits, representing block size 16B/32B/64B/128B/256B/512B/1024B. Considering the IE size constrained by [IEEE802154], 16B/32B/64B block size will be used. Invalid block size values will cause the packet to be dropped quietly.
- o Whether more blocks are following (M): 1 bit;
- o The relative number of the block (NUM): 12 bits, within a sequence of blocks with the given size. NUM is 4bits or 12bits, or 20bits

#### 4.3. Management Interface Protocol

Management and MIB handling is handled by the protocol specification defined in [I-D.ietf-6tisch-coap].

#### 4.4. Negotiation interface protocol

The negotiation protocol is used by neighbor nodes to agree at what slotOffset/channelOffset to add/remove softcells. It uses a Uri-Path option to identify the target resource (i.e the negotiation interface of the neighbor).

The example below illustrates the use of this negotiation interface. It assumes the RPC softcell-negotiation is at Uri-Path "6t/6/ng".



```

nodeA    nodeB
|         |
+-----> | IEEE802.15.4e type: DATA
|         | CoAP Header: POST (T=CON)
|         | Uri-Path: "6t/6/ng"
|         | Payload: CBOR(
|         |     Opcode=RESERVATION,
|         |     RequiredBW,
|         |     SlotframeID,
|         |     TrackID,
|         |     NumOfCandidate,
|         |     CandidateList
|         | )
|         |
<-----+ | IEEE802.15.4e type: ACK
|         |
<-----+ | IEEE802.15.4e type: DATA
|         | CoAP Header: 2.04 Changed (T=CON, Code=2.04)
|         | Payload: CBOR(
|         |     NumOfCells,
|         |     ResultedCells
|         | )
|         |
+-----> | IEEE802.15.4e type: ACK
|         |

```

Node A send a CoAP POST request, using a confirmable message. Node B sends back a IEEE802.15.4e ACK to confirm reception. This layer 2 ACK does not give any indication about the correct handling of the command, or even about whether this command is well formatted and understood. Node B parses the CoAP IE, and if correct, calls the appropriate 6top command to allocate softcells. When the allocation is done, node B sends back a CoAP Response with the appropriate return code to node A as a IEEE802.15.4e data packet. The CoAP ACK MUST be piggybacked on the Response.

#### 4.5. Acknowledgement

For both non-fragmented CoAP message and fragmented CoAP message, an Acknowledgement message of CoAP is used. The Acknowledgement message of CoAP is inserted into a CoAP IE, which is carried in the Data Frame or Enhanced Acknowledgement frame of [IEEE802154e].

#### 4.6. Observe

The Observe mechanism is a option for 6top-to-6top communication. The Token in the CoAP message is used to bind Observe message and its Response messages.

## 5. Implementation Considerations

Similar to the formatting and the parser modules used by CoAP (Layer 5), a CoAP formatting and parser modules are present in the 6top sublayer.

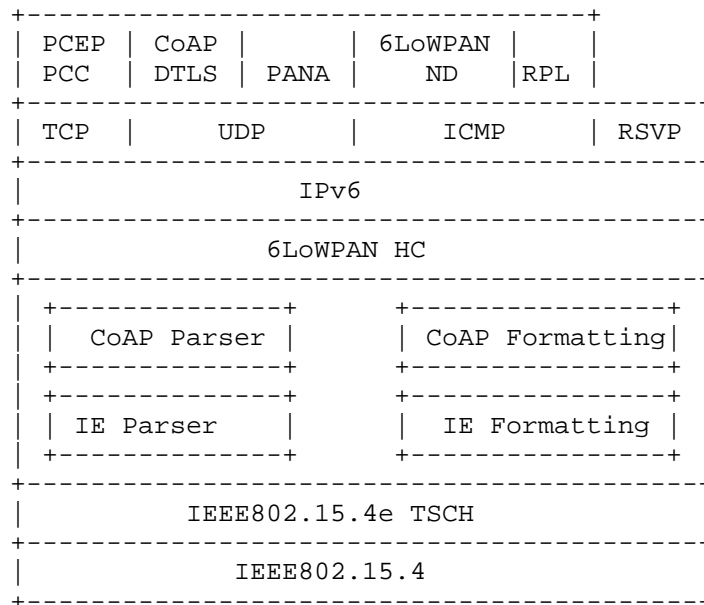


Figure 4

When the IE parser identifies a CoAP IE in the data packet, it passes the IE content (i.e. the fragmented CoAP message) to the CoAP Parser. The CoAP Parser then assembles those fragmented CoAP messages, and takes the appropriate action based on the CoAP Code, Uri-Path, and payload.

When a CoAP message is formatted, it MAY be fragmented, then passed to the IE Formatting module. The IE Formatting module puts those (possibly fragmented) CoAP message(s) into a CoAP IE and passes them to the IEEE802.15.4e TSCH layer as separate packets.

## 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

- [RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.
- [RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, June 2014.
- [I-D.ietf-core-block]  
Bormann, C. and Z. Shelby, "Blockwise transfers in CoAP", draft-ietf-core-block-14 (work in progress), October 2013.
- [I-D.ietf-6tisch-tsch]  
Watteyne, T., Palattella, M., and L. Grieco, "Using IEEE802.15.4e TSCH in an LLN context: Overview, Problem Statement and Goals", draft-ietf-6tisch-tsch-00 (work in progress), November 2013.
- [I-D.ietf-6tisch-architecture]  
Thubert, P., Watteyne, T., and R. Assimiti, "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-architecture-02 (work in progress), June 2014.
- [I-D.ietf-6tisch-terminology]  
Palattella, M., Thubert, P., Watteyne, T., and Q. Wang, "Terminology in IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-terminology-01 (work in progress), February 2014.
- [I-D.ietf-6tisch-minimal]  
Vilajosana, X. and K. Pister, "Minimal 6TiSCH Configuration", draft-ietf-6tisch-minimal-01 (work in progress), June 2014.
- [I-D.ietf-6tisch-6top-interface]  
Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top) Interface", draft-ietf-6tisch-6top-interface-00 (work in progress), March 2014.
- [I-D.ietf-6tisch-coap]  
Sudhaakar, R. and P. Zand, "6TiSCH Resource Management and Interaction using CoAP", draft-ietf-6tisch-coap-00 (work in progress), May 2014.

[I-D.wang-6tisch-6top-sublayer]

Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top)", draft-wang-6tisch-6top-sublayer-00 (work in progress), February 2014.

### 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.

[OpenWSN]

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 , August 2012.

[morell04label]

Morell, A., Vilajosana, X., Lopez-Vicario, J., and T. Watteyne, "Label Switching over IEEE802.15.4e Networks. Transactions on Emerging Telecommunications Technologies", June 2013.

### Authors' Addresses

Qin Wang (editor)  
Univ. of Sci. and Tech. Beijing  
30 Xueyuan Road  
Beijing, Hebei 100083  
China

Phone: +86 (10) 6233 4781  
Email: wangqin@ies.ustb.edu.cn

Xavier Vilajosana  
Universitat Oberta de Catalunya  
156 Rambla Poblenou  
Barcelona, Catalonia 08018  
Spain

Phone: +34 (646) 633 681  
Email: xvilajosana@uoc.edu

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

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

Raghuram S Sudhaakar  
Cisco Systems, Inc  
Building 24  
510 McCarthy Blvd  
San Jose 95135  
USA

Phone: +1 408 853 0844  
Email: rsudhaak@cisco.com

Pouria Zand  
University of Twente  
Department of Computer Science  
Zilverling Building  
Enschede 7522 NB  
The Netherlands

Phone: +31 619040718  
Email: p.zand@utwente.nl

6TiSCH  
Internet-Draft  
Intended status: Informational  
Expires: January 5, 2015

Q. Wang, Ed.  
Univ. of Sci. and Tech. Beijing  
X. Vilajosana  
Universitat Oberta de Catalunya  
T. Watteyne  
Linear Technology  
July 4, 2014

6TiSCH Operation Sublayer (6top)  
draft-wang-6tisch-6top-sublayer-01

Abstract

The recently published [IEEE802154e] standard formalizes the concept of link-layer resources in LLNs. Nodes are synchronized and follow a schedule. A cell in that schedule corresponds to an atomic link-layer resource, and can be allocated to any pair of neighbors in the network. This allows the schedule to be built to tightly match each node's bandwidth, latency and energy constraints. The [IEEE802154e] standard does not, however, present a mechanism to do so, as building and managing the schedule is out of scope of the standard. This document describes the 6TiSCH Operation Sublayer (6top) and the commands it provides to upper network layers such as RPL or GMPLS. The set of functionalities includes feedback metrics from cell states so network layers can take routing decisions, TSCH configuration and control procedures, and the support for decentralized, centralized or hybrid scheduling. In addition, 6top can be configured to enable packet switching at layer 2.5, analogous to GMPLS.

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 5, 2015.

#### Copyright Notice

Copyright (c) 2014 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 . . . . .	4
2. 6TiSCH Operation Sublayer (6top) Overview . . . . .	5
2.1. Cell Model . . . . .	7
2.1.1. hard cells . . . . .	8
2.1.2. soft cells . . . . .	8
2.2. Data Transfer Model . . . . .	8
3. 6top Commands . . . . .	11
3.1. Cell Commands . . . . .	13
3.1.1. CREATE.hardcell . . . . .	13
3.1.2. CREATE.softcell . . . . .	15
3.1.3. READ.cell . . . . .	16
3.1.4. UPDATE.cell . . . . .	17
3.1.5. DELETE.hardcell . . . . .	17
3.1.6. DELETE.softcell . . . . .	18
3.1.7. REALLOCATE.softcell . . . . .	19
3.2. Slotframe Commands . . . . .	19
3.2.1. CREATE.slotframe . . . . .	19
3.2.2. READ.slotframe . . . . .	20
3.2.3. UPDATE.slotframe . . . . .	20
3.2.4. DELETE.slotframe . . . . .	21
3.3. Monitoring Commands . . . . .	22
3.3.1. CONFIGURE.monitoring . . . . .	22
3.3.2. READ.monitoring.status . . . . .	22
3.4. Statistics Commands . . . . .	23

3.4.1.	CONFIGURE.statistics	23
3.4.2.	READ.statistics	23
3.4.3.	RESET.statistics	24
3.5.	Network Formation Commands	24
3.5.1.	CONFIGURE.eb	25
3.5.2.	READ.eb	25
3.6.	Time Source Neighbor Commands	26
3.6.1.	CONFIGURE.timesource	26
3.6.2.	READ.timesource	26
3.7.	Neighbor Commands	26
3.7.1.	CREATE.neighbor	27
3.7.2.	READ.all.neighbor	27
3.7.3.	READ.neighbor	27
3.7.4.	UPDATE.neighbor	27
3.7.5.	DELETE.neighbor	28
3.8.	Queueing Commands	28
3.8.1.	CREATE.queue	28
3.8.2.	READ.queue	28
3.8.3.	READ.queue.stats	29
3.8.4.	UPDATE.queue	29
3.8.5.	DELETE.queue	30
3.9.	Label Switching Commands	30
3.9.1.	LabelSwitching.map	30
3.9.2.	LabelSwitching.unmap	30
3.10.	Chunk Command	31
3.10.1.	Create.chunk	31
3.10.2.	READ.chunk	31
3.10.3.	Delete.chunk	32
3.11.	Chunk Cell Command	32
3.11.1.	CREATE.hardcell.fromchunk	32
3.11.2.	READ.chunkcell	33
3.11.3.	DELETE.hardcell.fromchunk	33
3.12.	Data Commands	34
3.12.1.	Send.data	34
3.12.2.	Receive.data	34
4.	6top Communication Protocol	35
4.1.	Message Formats	35
4.1.1.	Information Elements	35
4.1.2.	Packet Formats	43
4.2.	Time Sequences	48
4.2.1.	Network Formation	49
4.2.2.	Creating soft cells	50
4.2.3.	Deleting soft cells	51
4.2.4.	Maintaining soft cells	51
4.2.5.	Creating hard cells	51
4.2.6.	Deleting hard cells	52
5.	Statistics	52
5.1.	Statistics Metrics	52



5.2. Statistics Configuration . . . . .	53
6. Monitoring . . . . .	53
6.1. Monitor Configuration . . . . .	53
6.2. Actuation . . . . .	54
7. References . . . . .	54
7.1. Normative References . . . . .	54
7.2. Informative References . . . . .	54
7.3. External Informative References . . . . .	55
Authors' Addresses . . . . .	56

## 1. Introduction

As presented in [I-D.ietf-6tisch-tsch], the [IEEE802154e] standard defines the mechanisms for a TSCH node to communicate, given a schedule. It does not, however, define the mechanism to build and maintain the TSCH schedule, match that schedule to the multi-hop paths maintained by a network layer such as RPL or a 2.5 layer such as GMPLS, adapt the resources allocated between neighbor nodes to the data traffic flows, enforce a differentiated treatment for data generated at the application layer and signalling messages needed by 6LoWPAN and RPL to discover neighbors, react to topology changes, self-configure IP addresses, or manage keying material.

In a TSCH network, the MAC layer is not in charge of setting up the schedule that controls the connectivity graph of the network and the resources allocated to each node in that topology. This responsibility is left to the next-higher layer, defined in this document, called "6top".

This document describes the 6TiSCH Operation Sublayer (6top) and the main commands provided to upper network layers such as RPL or GMPLS. The set of functionalities include feedback metrics from cell state so the network layer can take routing decisions, TSCH configuration and control procedures, and support for the different scheduling mechanisms defined in [I-D.ietf-6tisch-architecture]. 6top addresses the set of functionalities described in [I-D.ietf-6tisch-tsch].

For example, network formation in a TSCH network involves the transmission of Enhanced Beacons (EB). EBs include information for joining nodes to be able to synchronize and set up an initial network topology. However, [IEEE802154e] does not specify how the period of EBs is configured, nor the rules for a node to select a particular node to join. 6top offers a set of commands so control mechanisms can be introduced on top of TSCH to configure nodes to join a specific node. Once a network is formed, 6top maintains the network's health, allowing for nodes to stay synchronized. It supplies mechanisms to manage each node's time source neighbor and configure the EB interval. Network layers running on top of 6top take advantage of

the TSCH MAC layer information so routing metrics, topological information, energy consumption and latency requirements can be adjusted to TSCH, and adapted to application requirements.

TSCH requires a mechanism to manage its schedule; 6top provides a set of commands for upper layers to set up specific schedules, either explicitly by detailing specific cell information, or by allowing 6top to establish a schedule given a bandwidth or latency requirement. 6top is designed to enable decentralized, centralized or hybrid scheduling solutions. 6top enables internal TSCH queuing configuration, size of buffers, packet priorities, transmission failure behavior, and defines mechanisms to encrypt and authenticate MAC slotframes.

As described in [label-switching-154e], due to the slotted nature of a TSCH network, it is possible to use a label switched architecture on top of TSCH cells. As a cell belongs to a specific track, a label header is not needed at each packet; the input cell (or bundle) and the output cell (or bundle) uniquely identify the data flow. The 6top sublayer provides operations to manage the cell mappings.

## 2. 6TiSCH Operation Sublayer (6top) Overview

6top is a sublayer which is the next-higher layer for TSCH (Figure 1), which architecture is detailed in [I-D.ietf-6tisch-architecture], and generic data model is detailed in [I-D.ietf-6tisch-6top-interface]. 6top offers both management and data interfaces to an upper layer. It includes monitoring and statistics collection, both of which are configurable through the management interface.

## Protocol Stack

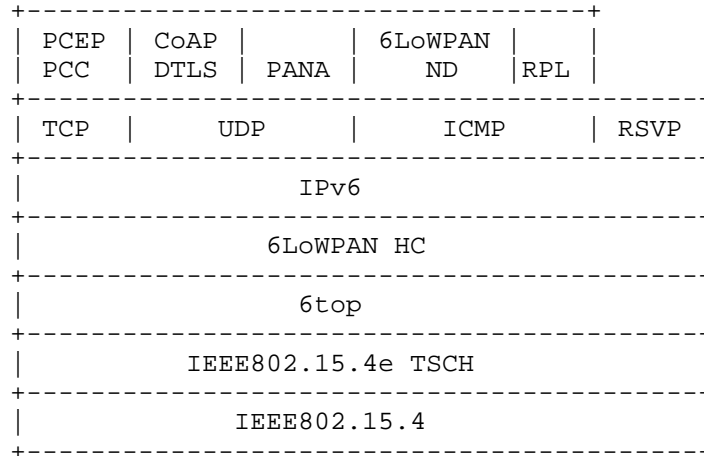


Figure 1

6top distinguishes between hard cells and soft cells. It therefore requires an extra flag to all cells in the TSCH schedule, as detailed in Section 2.1.

When a higher layer gives 6top a 6LoWPAN packet for transmission, 6top maps it to the appropriate outgoing priority-based queue, as detailed in Section 2.2.

All 6top commands of the management and data interfaces are detailed in Section 3. This set of commands is designed to support decentralized, centralized and hybrid scheduling solutions. They form a conceptual interface an upper layer can use; implementations can use this set of commands, or any equivalent alternative.

6top defines TSCH Information Elements (IEs) for neighbors nodes to negotiate scheduling cells in the TSCH schedule. The format of those IEs is given in Section 4.1. Example data exchanges between neighbor nodes are given in Section 4.2.

Section 5 defines how 6top gathers statistics (e.g. link quality, energy level, queue usage), and what commands an upper layer can use to configure and retrieve those statistics.

6top can be configured to monitor the cells it has scheduled in order to detect cells with poor performance. It can automatically re-allocate those cells inside the TSCH schedule. This behavior is described in Section 6

## 2.1. Cell Model

[IEEE802154e] defines a set of options attached to each cell. A cell can be a Transmit cell, a Receive cell, a Shared cell or a Timekeeping cell. These options are not exclusive, as a cell can be qualified with more than one of them. The MLME-SET-LINK.request command defined in [IEEE802154e] uses a linkOptions bitmap to specify the options of a cell. Acceptable values are:

b0 = Transmit

b1 = Receive

b2 = Shared

b3 = Timekeeping

b4-b7 = Reserved

Only Transmit cells can also be marked as Shared cells. When the shared bit is set, a back-off procedure is applied to handle collisions. Shared behavior does not apply to Receive cells.

6top allows an upper layer to schedule a cell at a specific slotOffset and channelOffset, in a specific slotframe.

In addition, 6top allows an upper layer to schedule a certain amount of bandwidth to a neighbor, without having to specify the exact slotOffset and channelOffset of the corresponding cell(s). Once bandwidth is reserved, 6top is in charge of ensuring that this requirement is continuously satisfied. 6top dynamically reallocates cells if needed, and over-provisions if required.

6top allows an upper layer to associate a cell with a specific track by using a TrackID. A TrackID is a tuple (TrackOwnerAddr, InstanceID). TrackOwnerAddr is the address of the node which initiates the process of creating the track, i.e. the owner of the track. InstanceID is an instance identifier given by the owner of the track. InstanceID comes from the upper layer; it could for example be the local instance ID defined in RPL.

If the TrackID is set to (0,0), the cell can be used by the best-effort QoS configuration or as a Shared cell. If the TrackID is not set to (0,0), i.e. the cell belongs to a specific track, the cell MUST not be set as Shared cell.

6top allows an upper layer to ask a node to manage a portion of a slotframe, called a chunk. Chunks can be delegated explicitly by the

PCE to a node, or claimed automatically by any node that participates to the distributed cell scheduling process. The cells in a chunk can be appropriated by the node, i.e. the node is in charge of managing the chunk.

Given this mechanism, 6top defines hard cells (which have been requested specifically) and soft cells (which can be reallocated dynamically). The hard/soft flag is introduced by the 6top sublayer named as CellType (0: soft cell, 1: hard cell). This option is mandatory; all cells are either hard or soft.

#### 2.1.1. hard cells

A hard cell is a cell that cannot be dynamically reallocated by 6top. A hard cell is uniquely identified by the following tuple:

slotframe ID: ID of the slotframe this cell is part of.

slotOffset: the slotOffset for the cell.

channelOffset: the channelOffset for the cell.

LinkOption bitmap: bitmap as defined in [IEEE802154].

CellType: MUST be set to 1.

#### 2.1.2. soft cells

A soft cell is a cell that can be reallocated by 6top dynamically. The CellType MUST be set to 0. This cell is installed by 6top given a specific bandwidth requirement. Soft cells are installed through the soft cell negotiation procedure described in Section 4.2.

### 2.2. Data Transfer Model

The TSCH MAC layer is decoupled from the upper layer; the interaction between the upper layer and TSCH is asynchronous. This means that the MAC layer executes a schedule and checks at each timeslot according to the type of cell (i.e Transmit, Shared or Receive), whether there is something to send or receive. If that is the case, the packet is transmitted and the MAC layer continues its operation. When an upper layer sends a packet, this packet is pushed into a queue waiting for the MAC layer to read it and send it in a particular timeslot according to its destination and priority. 6top provides a set of queue management operations which enable upper layers to create different queues and set their priorities. This allows different classes of traffic to be handled by the forwarding

plane by inserting a packet into the queue appropriate for its priority.

A 6top implementation MUST provide at least a Broadcast Queue and a Transmit Queue. The Broadcast Queue is associated with cells with LinkType=ADVERTISING in the sender's schedule, and LinkOption="Receive" and "Timekeeping" in all its neighbors' schedule. For example, NodeA uses slotOffset=5 and channelOffset=12 as Broadcast cell to its neighbors NodeB and NodeC. Then, in the schedule of NodeA the cell will be featured with neighbor address is Broadcast address, LinkType=ADVERTISING; and in the schedules of both nodeB and nodeC the cell will be featured with nodeA address as neighbor address, and LinkOption="Receive" and "Timekeeping", which ensure nodeB and nodeC will be active at least one time in the cell to receive broadcast packet during a Timekeeping period. A Transmit Queue is associated with the dedicated Transmit cells or Shared Cells.

Data Communication Commands (Section 3.12) can be used to send control messages and data messages. The operation is used to insert a message into a specific queue.

For example, a configuration can include two Broadcast Queues with priority High and Low, and three Transmit Queues with priority High, Mid, and Low.

When DestAddr is the broadcast address, its related MAC layer packets will be pushed into the Broadcast Queue with the corresponding priority. 6top is responsible for feeding these packets into broadcast cells.

When DestAddr is a unicast address, its related MAC layer packets will be pushed into the Transmit Queue with the corresponding priority. 6top is responsible for feeding these packets into Transmit or Shared Cells.

The QoS policy enforced by 6top is out of scope. As an example, packets in higher priority queues could be transmitted before the packets in lower priority queue. As a result, when there is an available broadcast/unicast cell, 6top checks the broadcast/unicast queue with higher priority first. 6top continues this search until it finds a broadcast/unicast packet, or finds that all of broadcast/unicast queues are empty.

Figure 2 shows how 6top shapes data from the upper layer (e.g., RPL, 6LoWPAN), and feeds it to TSCH. The properties associated with a packet/fragment from the upper layer includes the next hop neighbor (DestAddr), the packet priority, and TrackID(s).

## 6top Data Transfer Model

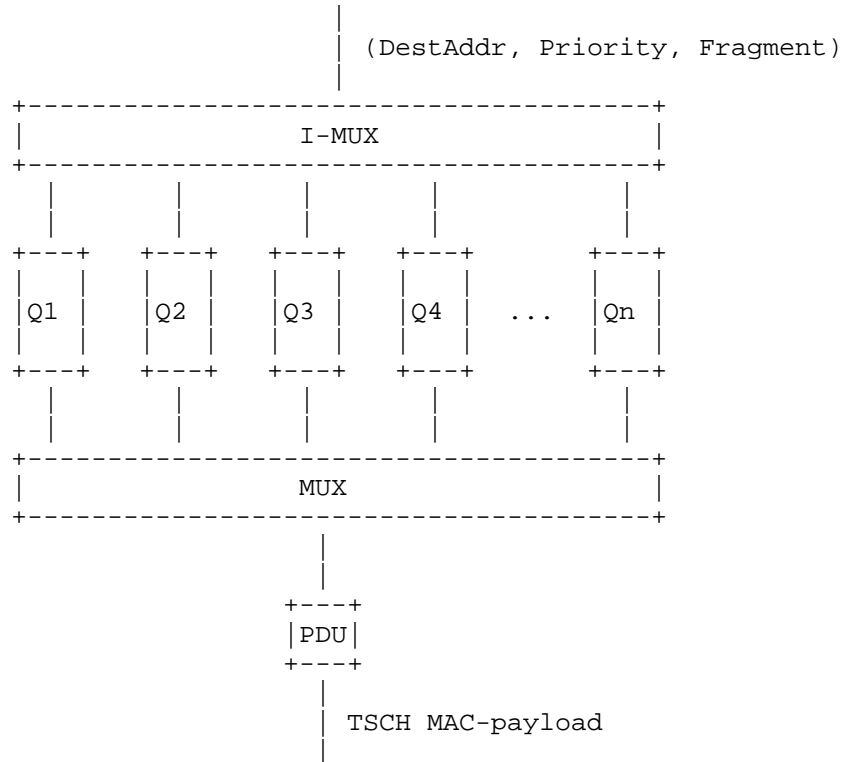


Figure 2

In Figure 2,  $Q_i$  represents a queue, which is either broadcast or unicast, and is assigned a priority. The number of queues is configurable. The relationship between queues and tracks is configurable. For example, for a given queue, only one specific track can be used, all of the tracks can be used, or a subset of the tracks can be used.

When 6top receives a packet to transmit through a `Send.data` command (Section 3.12), the `I-MUX` module selects a queue in which to insert it. If the packet's destination address is a unicast (resp. broadcast) address, it is inserted into a unicast (resp. broadcast) queue.

The `MUX` module is invoked at each scheduled transmit cell by TSCH. When invoked, the `MUX` module goes through the queues, looking for the best matching frame to send. If it finds a frame, it hands it over

to TSCH for transmission. If the next active cell is a broadcast cell, it selects a fragment only from broadcast queues.

How the MUX module selects the best frame is configurable. The following rules are a typical example:

The frame's layer 2 destination address MUST match the neighbor address associated with the transmit cell.

If the transmit cell is associated with a specific track, the frames in the queue corresponding to the TrackID have the highest priority.

If the transmit cell is not associated with a specific track, i.e., TrackID=(0,0), frames from a queue with a higher priority MUST be sent before frames from a queue with a lower priority.

Further rules can be configured to satisfy specific QoS requirements.

### 3. 6top Commands

6top provides a set of commands as the interface with the higher layer. Most of these commands are related to the configuration of slotframes, cells and scheduling information. 6top also provides an interface allowing an upper layer to retrieve status information and statistics. The management commands provided by 6top are listed below. Note that this set defines a conceptual interface only; an implementation can choose to use this exact set of commands, or any equivalent alternative.

CREATE.hardcell: Section 3.1.1

CREATE.softcell: Section 3.1.2

READ.cell: Section 3.1.3

UPDATE.cell: Section 3.1.4

DELETE.hardcell: Section 3.1.5

DELETE.softcell: Section 3.1.6

REALLOCATE.softcell: Section 3.1.7

CREATE.slotframe: Section 3.2.1

READ.slotframe: Section 3.2.2



UPDATE.slotframe: Section 3.2.3

DELETE.slotframe: Section 3.2.4

CONFIGURE.monitoring: Section 3.3.1

READ.monitoring: Section 3.3.2

CONFIGURE.statistics: Section 3.4.1

READ.statistics: Section 3.4.2

RESET.statistics: Section 3.4.3

CONFIGURE.eb: Section 3.5.1

READ.eb: Section 3.5.2

CONFIGURE.timesource: Section 3.6.1

READ.timesource: Section 3.6.2

CREATE.neighbor: Section 3.7.1

READ.all.neighbor: Section 3.7.2

READ.neighbor: Section 3.7.3

UPDATE.neighbor: Section 3.7.4

DELETE.neighbor: Section 3.7.5

CREATE.queue: Section 3.8.1

READ.queue: Section 3.8.2

READ.queue.stats: Section 3.8.3

UPDATE.queue: Section 3.8.4

DELETE.queue: Section 3.8.5

LabelSwitching.map: Section 3.9.1

LabelSwitching.unmap: Section 3.9.2

CREATE.chunk: Section 3.10.1

READ.chunk: Section 3.10.2

DELETE.chunk: Section 3.10.3

CREATE.hardcell.fromchunk: Section 3.11.1

READ.chunkcell: Section 3.11.2

DELETE.hardcell.fromchunk: Section 3.11.3

Besides management commands, 6top provides the following data commands:

Send.data: Section 3.12.1

Receive.data: Section 3.12.2

In addition, 6top offers a delegation interface allowing an upper layer to configure TSCH. 6top only delegates the functionalities to the MAC security services. In other words, 6top allows an upper layer to access the security PIB (Table 60, Table 61, Table 63 in [IEEE802154]) by using MLME-GET/MLME-SET primitives defined in [IEEE802154].

### 3.1. Cell Commands

6top provides the following commands to manage TSCH cells.

#### 3.1.1. CREATE.hardcell

Creates one or more hard cells in the schedule. Fails if the cell already exists. A cell is uniquely identified by the tuple (slotframe ID, slotOffset, channelOffset).

To create a hard cell, the upper layer specifies:

slotframe ID: ID of the slotframe this timeslot will be scheduled in.

slotOffset: the slotOffset for the cell.

channelOffset: channelOffset for the cell.

LinkOption bitmap: bitmap as defined in [IEEE802154e]

LinkType : as defined in section 6.2.19.3 of [IEEE802154e].

CellType: as defined in Section 2.1

target node address: the address of that node to communicate with over this cell. In case of broadcast cells this is the broadcast address.

TrackID: ID of the track the cell will belong to.

6top schedules the cell and marks it as a hard cell, indicating that it cannot reschedule this cell. The return value is CellID and the created cell is also filled in CellList ([I-D.ietf-6tisch-6top-interface]).

The interaction between 6top and MAC layer caused by CREATE.hardcell is as follows.

Firstly, 6top calls the primitive MLME-SET-LINK.request defined in section 6.2.19.3 of [IEEE802154e]. The primitive parameters are set as follows.

MLME-SET-LINK.request parameter	set by 6top
operation	ADD-LINK
LinkHandle	CellID
slotframeHandle	slotframe ID
timeslot	slotOffset
channelOffset	channelOffset
LinkOptions	LinkOption bitmap
LinkType	LinkType
nodeAddr	target node address

Secondly, if the status from MLME-SET-LINK.confirm defined in section 6.2.19.4 of [IEEE802154e] is SUCCESS, then add the LinkHandle to the BundleList specified by TrackID, and confirm to upper layer with status = SUCCESS; otherwise, confirm to upper layer with status = FAIL.

### 3.1.2. CREATE.softcell

To create soft cell(s), the upper layer specifies:

slotframe ID: ID of the slotframe the cell(s) will be scheduled in

number of cells: the required number of soft cells.

LinkOption bitmap: bitmap as defined in [IEEE802154e]

CellType: as defined in Section 2.1

target node address: the address of the node to communicate with over the cell(s). In case of broadcast cells this is the broadcast address.

TrackID: ID of the track the cell(s) will belong to.

QoS level: the cell redundancy policy. The policy can be for example STRICT, BEST\_EFFORT, etc.

6top is responsible for picking the exact slotOffset and channelOffset in the schedule, and ensure that the target node choose the same cell and TrackID. 6top marks these cells as soft cell, indicating that it will continuously monitor their performance and reschedule if needed. The return value is CellID, and the created cell is also filled in CellList ([I-D.ietf-6tisch-6top-interface]).

6top deals with the allocation process by negotiation with the target node. The command returns the number and the list of created cells defined by (slotframe ID, slotOffset, channelOffset). The number of crated cells is less than the required number of cells if the required number of cells is higher than the available number of cells in the schedule. The number of created cells equals to zero if the negotiation with the target node fails. The number of created cells equals to zero if the CellType bitmap indicates that the cell(s) MUST be Hard.

The interaction between 6top and TSCH happens on both sides described as follows.

For example, after negotiation, node A and node B find a specific cell, slotOffset=10, channelOffset=12, as a Tx cell and Rx cell, respectively, then the 6top in node A and node B will call the primitive MLME-SET-LINK.request defined in section 6.2.19.3 of [IEEE802154e], respectively. The primitive parameters are set in node A and node B as follows.

MLME-SET-LINK.request parameter	set by A's 6top	set by B's top
operation	ADD-LINK	ADD-LINK
LinkHandle	CellID	CellID
slotframeHandle	slotframe ID	slotframe ID
timeslot	10	10
channelOffset	12	12
LinkOptions	Tx	Rx
LinkType	NORMAL	NORMAL
nodeAddr	Node A	Node B

If the Status from MLME-SET-LINK.confirm defined in section 6.2.19.4 of [IEEE802154e], 6top will notify upper layer failure.

### 3.1.3. READ.cell

Given a (slotframe ID, slotOffset, channelOffset), retrieves the cell information. Fails if the cell does not exist. The returned information contains:

slotframe ID: ID of the slotframe where this cell is installed.

slotOffset: the slotOffset for the cell.

channelOffset: the selected channelOffset for the cell.

LinkOption bitmap: bitmap as defined in [IEEE802154e]

CellType: as defined in Section 2.1

target node address: the target address of that cell. In case of broadcast cells this is the broadcast address.

TrackID: ID of the track the cell will belong to.

NumOfStatistics: Number of elements in the following list of tuple (StatisticsMetricID and StatisticsValue)

list of (StatisticsMetricID, StatisticsValue):  
StatisticsMetricID is the index to Statistics Metric defined in Section 3.4, StatisticsValue is the value corresponding to the metric indexed by StatisticsMetricID

A read command can be issued for any cell, hard or soft. 6top gets cell information from CellList ([I-D.ietf-6tisch-6top-interface]).

#### 3.1.4. UPDATE.cell

Update a hard cell, i.e., re-allocate it to a different slotOffset and/or channelOffset. Fails if the cell does not exist. Requires both old (slotframe ID, slotOffset, channelOffset) and new (slotframe ID, slotOffset, channelOffset) as parameters. And, the type of cell, target node address and TrackID are the fields that cannot be updated. Soft cells MUST NOT be updated by the UPDATE.cell command. REALLOCATE.softcell (Section 3.1.7) MUST be used instead.

It causes a old cell being removed and a new cell being created.

#### 3.1.5. DELETE.hardcell

To remove a hard cell, the upper layer specifies:

slotframe ID: the ID of the slotframe where this cell is installed.

slotOffset: the slotOffset for the cell.

channelOffset: the selected channelOffset for the cell.

LinkOption bitmap: bitmap as defined in [IEEE802154e]

LinkType : as defined in section 6.2.19.3 of [IEEE802154e].

CellType: as defined in Section 2.1

target node address: the target address of that cell. In case of broadcast cells this is the broadcast address.

TrackID: ID of the track the cell will belong to.

This removes the hard cell from the node's schedule, from CellList ([I-D.ietf-6tisch-6top-interface]) as well.

The interaction between 6top and MAC layer caused by DELETE.hardcell is as follows.

Firstly, 6top calls the primitive MLME-SET-LINK.request defined in section 6.2.19.3 of [IEEE802154e]. The primitive parameters are set as follows.

MLME-SET-LINK.request parameter	set by 6top
operation	DELETE-LINK
LinkHandle	CellID
slotframeHandle	slotframe ID
timeslot	slotOffset
channelOffset	channelOffset
LinkOptions	LinkOption bitmap
LinkType	LinkType
nodeAddr	target node address

Secondly, if the status from MLME-SET-LINK.confirm defined in section 6.2.19.4 of [IEEE802154e] is SUCCESS, then remove the LinkHandle from its BundleList specified by TrackID, and confirm to upper layer with status = SUCCESS; otherwise, confirm to upper layer with status = FAIL.

### 3.1.6. DELETE.softcell

To remove a (number of) soft cell(s), the upper layer specifies:

slotframe ID: ID of the slotframe where this cell is installed.

number of cells: the number of cells to be removed

LinkOption bitmap: bitmap as defined in [IEEE802154e]

CellType: as defined in Section 2.1

target node address: the target address of that cell. In case of broadcast cells this is the broadcast address.

TrackID: ID of the track the cell will belong to.

In the case a soft cell wants to be re-allocated from the allocated cell so a hard cell can be installed instead, the `REALLOCATE.softcell` (Section 3.1.7) MUST be used.

After the pair of nodes figure out the specific cell(s) to be removed, the interaction between 6top and TSCH on both sides will be similar to that caused by `DELETE.hardcell`, except `LinkType` should be set to `NORMAL`.

### 3.1.7. `REALLOCATE.softcell`

To force a re-allocation of a soft cell, the upper layer specifies:

slotframe ID: ID of the slotframe where the cell is allocated.

slotOffset: the slotOffset for that cell.

channelOffset: the channelOffset for that cell.

The reallocated cell will be installed in a different slotOffset, channelOffset but slotframe and TrackID remain the same. Hard cells MUST NOT be reallocated.

The interaction between 6top and TSCH caused by this command includes that described in Section 3.1.6 and Section 3.1.2.

## 3.2. Slotframe Commands

6top provides the following commands to manage TSCH slotframes.

### 3.2.1. `CREATE.slotframe`

Creates a new slotframe. The command requires:

slotframe ID: unique identifier of the slotframe, corresponding to its priority.

number of timeslots: the required number of timeslots in the slotframe.

Fails if the number of required timeslots is less than zero.

The interaction between 6top and MAC layer caused by `CREATE.slotframe` is as follows.

Firstly, 6top calls the primitive `MLME-SET-SLOTFRAME.request` defined in section 6.2.19.1 of [IEEE802154e]. The primitive parameters are set as follows.



MLME-SET-SLOTFRAME.request parameter	set by 6top
slotframeHandle	slotframe ID
operation	ADD
size	number of timeslot

Secondly, if the status from MLME-SET-SLOTFRAME.confirm defined in section 6.2.19.2 of [IEEE802154e] is SUCCESS, then confirms to upper layer with status = SUCCESS; otherwise, confirm to upper layer with status = FAIL.

### 3.2.2. READ.slotframe

Returns the information of a slotframe given its slotframe ID. The command returns:

slotframe ID: ID of the slotframe. (SlotFrameHandle)

number of timeslots: the number of timeslots in the slotframe.

Fails if the slotframe ID does not exist.

### 3.2.3. UPDATE.slotframe

Change the number of timeslots in a slotframe. The command requires:

slotframe ID: ID of the slotframe.

number of timeslots: the number of timeslots to be updated.

Fails if the number of required timeslots is less than zero. Fails if the slotframe ID does not exist.

The interaction between 6top and MAC layer caused by UPDATE.slotframe is as follows.

Firstly, 6top calls the primitive MLME-SET-SLOTFRAME.request defined in section 6.2.19.1 of [IEEE802154e]. The primitive parameters are set as follows.

MLME-SET-SLOTFRAME.request parameter	set by 6top
slotframeHandle	slotframe ID
operation	MODIFY
size	number of timeslot

Secondly, if the status from MLME-SET-SLOTFRAME.confirm defined in section 6.2.19.2 of [IEEE802154e] is SUCCESS, then confirms to upper layer with status = SUCCESS; otherwise, confirm to upper layer with status = FAIL.

#### 3.2.4. DELETE.slotframe

Deletes a slotframe. The command requires:

slotframe ID: ID of the slotframe.

number of timeslot: the number of timeslots in the slotframe.

Fails if the slotframe ID does not exist.

The interaction between 6top and MAC layer caused by DELETE.slotframe is as follows.

Firstly, 6top calls the primitive MLME-SET-SLOTFRAME.request defined in section 6.2.19.1 of [IEEE802154e]. The primitive parameters are set as follows.

MLME-SET-SLOTFRAME.request parameter	set by 6top
slotframeHandle	slotframe ID
operation	DELETE
size	number of timeslot

Secondly, if the status from MLME-SET-SLOTFRAME.confirm defined in section 6.2.19.2 of [IEEE802154e] is SUCCESS, then confirms to upper layer with status = SUCCESS; otherwise, confirm to upper layer with status = FAIL.

### 3.3. Monitoring Commands

Monitoring commands provide the means for upper layers to configure whether 6top must ensure the required bandwidth. This procedure is achieved through overprovisioning according to cell status feedback. Monitoring is also in charge of reallocating soft cells that are under the required QoS.

#### 3.3.1. CONFIGURE.monitoring

Configures the level of QoS the Monitoring process MUST enforce. The command requires:

slotframe ID: ID of the slotframe.

target node address: the target neighbor address.

enforce policy: The policy used to enforce the QoS requirements. Can be for example DISABLE, BEST\_EFFORT, STRICT, OVER-PROVISION, etc.

Fails if the slotframe ID does not exist.

#### 3.3.2. READ.monitoring.status

Reads the current Monitoring status. Requires the following parameters.

slotframe ID: the ID of the slotframe.

target node address: the target neighbor address.

Returns the QoS levels for that Target node on that slotframe.

allocated\_hard: Number of hard cells allocated.

allocated\_soft: Number of soft cells allocated.

provisioned: the extra provisioned cells. 0 if CONFIGURE.qos enforce is DISABLE.

QoS: the current QoS. Including overprovisioned cells, i.e what bandwidth is being obtained including the overprovisioned cells.

RQoS: the real QoS without provisioned cells. What is the actual bandwidth without taking into account the overprovisioned cells.

Fails if the slotframe ID does not exist.

### 3.4. Statistics Commands

6top keeps track of TSCH statistics for upper layers to adapt correctly to medium changes. The exact metrics for statistics are out of scope but the present commands SHOULD be used to configure and read monitored information regardless of the specific metric.

#### 3.4.1. CONFIGURE.statistics

Configures Statistics process. The command requires:

slotframe ID: ID of the slotframe. If empty monitors all slotframe IDs

slotOffset: specific slotOffset to be monitored. If empty all timeslots are monitored

channelOffset: specific channelOffset to be monitored. If empty all channels are monitored.

target node address: the target neighbor address. If empty, all neighbor nodes are monitored.

metric: metric to be monitored. This MAY be PDR, ETX, queuing statistics, energy-related metrics, etc.)

window: time window to be considered for the calculations. If 0 all historical data is considered.

enable: Enables statistics or disables them.

Fails if the slotframe ID does not exist. The statistics service can be configured to retrieve statistics at different levels. For example to aggregate information by slotframe ID, or to retrieve statistics for a particular timeslot, etc. The CONFIGURE.statistics enables flexible configuration and supports empty parameters that will force 6top to conduct statistics on all members of that dimension. For example, if ChannelOffset is empty and metric is set as PDR, then, 6top will conduct the statistics of PDR on all of channels.

#### 3.4.2. READ.statistics

Reads a metric for the specified dimension. Information is aggregated according to the parameters. The command requires:

slotframe ID: ID of the slotframe. If empty aggregates information of all slotframe IDs

slotOffset: the specific slotOffset for which the information is required. If empty all timeslots are aggregated

channelOffset: the specific channelOffset for which the information is required. If empty all channels are aggregated.

target node address: the target neighbor address. If empty all neighbor addresses are aggregated.

metric: metric to be read.

Returns the value for the requested metric.

Fails if empty metric or metric does not exists.

### 3.4.3. RESET.statistics

Resets the gathered statistics. The command requires:

slotframe ID: ID of the slotframe. If empty resets the information of all slotframe IDs

slotOffset: the specific slotOffset for which the information wants to be reset. If empty statistics from all timeslots are reset

channelOffset: the specific channelOffset for which the information wants to be reset. If empty all statistics for all channels are reset.

target node address: the target neighbor address. If empty all neighbor addresses are aggregated.

metric: metric to be reset.

Fails if empty metric or metric does not exists.

### 3.5. Network Formation Commands

EBs need to be configured, including their transmission period, the slotOffset and channelOffset that they SHOULD be sent on, and the join priority they contain. The parameters for that command are optional and enable flexible configuration of EBs. If slotframe ID is specified, the EBs will be configured to use that specific slotframe; if not, they will use the first slotframe where the

configured slotOffset is allocated. The slotOffset enforces the EB to a specific timeslot. In case slotOffset parameter is not present, the EB is sent in the first available transmit timeslot. In case channelOffset parameter is not set, the EB is configured to use the first available channel.

### 3.5.1. CONFIGURE.eb

Configures EBs. The command requires:

slotframe ID: ID of the slotframe where the EBs MUST be sent. Zero if any slotframe can be used.

slotOffset: the slotOffset where the EBs MUST be sent. Zero if any timeslot can be used.

channelOffset: the channelOffset where the EBs MUST be sent. Zero if any channelOffset can be used.

period: the EBs period, in seconds.

Expiration: when the EBs periodicity will stop. If Zero the period never stops.

priority: the joining priority model that will be used for advertisement. Joining priority MAY be for example SAME\_AS\_PARENT, RANDOM, BEST\_PARENT+1 or DAGRANK(rank) as deccribed in in [I-D.ietf-6tisch-minimal].

Fails if the tuple (slotframe ID, slotOffset, channelOffset) is already scheduled.

### 3.5.2. READ.eb

Reads the EBs configuration. No parameters are required.

Returns the current EBs configuration for that slotframe, which contains:

slotframe ID: the slotframe where the EB is being sent.

slotOffset: the slotOffset where the EBs is being sent.

channelOffset: the channelOffset the EBs is being sent on.

period: the EBs period.

Expiration: when the EBs periodicity stops. If 0 the period never stops.

priority: the joining priority that this node advertises.

Fails if the slotframe ID does not exist.

### 3.6. Time Source Neighbor Commands

Commands to select time source neighbors.

#### 3.6.1. CONFIGURE.timesource

Configures the Time Source Neighbor selection process. More than one time source neighbor can be selected. The command requires:

selection policy: The policy used to select the time source neighbor. The policy MAY be for example ALL\_PARENTS, BEST\_CONNECTED, LOWEST\_JOIN\_PRIORITY, etc.

Fails if any of the time source neighbors do not exist or it is not reachable.

#### 3.6.2. READ.timesource

Retrieves information about the time source neighbors of that node. The command does not require any parameter.

Returns the following information for each of the time sources:

target node: address of the time source neighbor.

statistics: includes for example minimum, maximum, average time correction for that time source neighbor

policy: the used policy

Fails if the slotframe ID or no time source neighbors exist.

### 3.7. Neighbor Commands

Commands to manage neighbor table. The commands SHOULD be used by the upper layer to query the neighbor related information and by the lower layer to keep track of neighbors information.

### 3.7.1. CREATE.neighbor

Creates an entry for a neighbor in the neighbor table.

neighbor address: The address of the neighbor.

neighbor stats: for example, RSSI of the last received packet from that neighbor, ASN when that neighbor has been added, etc.

Returns whether the neighbor is created or not.

### 3.7.2. READ.all.neighbor

Returns the list of neighbors of that node. Fails if empty. For each neighbor in the list it returns:

neighbor address: The address of the neighbor.

neighbor stats: for example, RSSI of the last received packet from that neighbor, ASN when that neighbor has been added, packets received from that neighbor, packets sent to it, etc.

### 3.7.3. READ.neighbor

Returns the information of a specific neighbor of that node specified by its neighbor address. Fails if it does not exist. For that neighbor it returns:

neighbor address: The address of the neighbor.

neighbor stats: for example, RSSI of the last received packet from that neighbor, ASN when that neighbor has been added, packets received from that neighbor, packets sent to it, etc.

### 3.7.4. UPDATE.neighbor

Updates an entry for a neighbor in the neighbor table. Fails if the neighbor does not exist. Updates stats parameters. Requires:

neighbor address: The address of the neighbor.

neighbor stats: for example, RSSI of the last received packet from that neighbor, ASN when that neighbor has been added, etc.

Returns whether the neighbor is updated or not.



### 3.7.5. DELETE.neighbor

Deletes a neighbor given its address. Fails if the neighbor does not exist.

## 3.8. Queueing Commands

Queues need to be configured. This includes queue length, retransmission policy, discarding of packets, etc.

### 3.8.1. CREATE.queue

Creates and Configures Queues. The command SHOULD be applied for each required queue. The command requires:

txqlength: the desired transmission queue length.

rxqlength: the desired reception queue length.

numrtx: number of allowed retransmissions.

age: discard packet according to its age on the queue. 0 if no discards are allowed.

rtxbackoff: retransmission backoff in number of slotframes. 0 if next available timeslot wants to be used.

statswindow: window of time used to compute statistics.

queue priority: the priority of this queue.

TrackIDs: a set of TrackIDs. While it is empty, no specific track is associated with the queue

Returns the queue ID.

### 3.8.2. READ.queue

Reads the queue configuration. Requires the queue ID.

The command returns:

txqlength: the transmission queue length.

rxqlength: the reception queue length.

numrtx: number of allowed retransmissions.

age: maximum age of a packet before being discarded. 0 if no discards are allowed.

rtxbackoff: retransmission backoff in number of slotframes. 0 if next available timeslot is used.

### 3.8.3. READ.queue.stats

Reads the queue stats. Requires queue ID.

The command returns:

txqlengthstats: average, maximum, minimum length of the transmission queue.

rxqlengthstats: average, maximum, minimum length of the reception queue.

numrtxstats: average, maximum, minimum number of retransmissions.

agestats: average, maximum, minimum age of a packet in the queue.

rtxbackoffstats: average, maximum, minimum retransmission backoff.

queue priority: the priority of this queue.

TrackIDs: a set of TrackIDs.

### 3.8.4. UPDATE.queue

Update a Queue. The command requires:

queueid: the queue ID.

txqlength: the desired transmission queue length.

rxqlength: the desired reception queue length.

numrtx: number of allowed retransmissions.

age: discard packet according to its age on the queue. 0 if no discards are allowed.

rtxbackoff: retransmission backoff in number of slotframes. 0 if next available timeslot wants to be used.

statswindow: window of time used to compute stats.

queue priority: the desired priority of this queue.

TrackIDs: the desired set of TrackIDs.

### 3.8.5. DELETE.queue

Deletes a Queue. The command requires the queue ID. All packets in the queue are discarded and the queue is deleted.

## 3.9. Label Switching Commands

6top is responsible for maintaining the mapping of input cells and output cells in the same track in a particular node. By keeping that mapping, layer 3 routing can be avoided as packets are forwarded by the 6top sublayer according to the input cells they were received on. The selected output cell is one of the cells that forward the packet to the subsequent hop in the track.

### 3.9.1. LabelSwitching.map

The command used by an upper layer to map an input cell or a bundle of input cells to an output cell or a bundle of output cells. 6top stores this mapping and makes sure that the packets are forwarded at the specific output cell/bundle. Label Switching is enabled by the specified bundle as soon as the mapping is installed.

The required parameters are:

input cells: list of input cells (one or more cells in a bundle). Each input cells is described by a unique tuple (slotOffset, channelOffset, destination address).

output cells: list of output cells (one or more cells in a bundle). Each output cells is described by a unique tuple (slotOffset, channelOffset, destination address).

load balancing policy: A policy for load balance cell usage. The policy is out of scope, however an example can be use ROUND ROBIN policy within the cells of the same bundle.

### 3.9.2. LabelSwitching.unmap

The command used by upper layers to unmap one input cell or a bundle of input cells to an output cell or a bundle of output cells. The mapping is removed from the state kept by 6top.

The required parameters are:

input cells: list of input cells (one or more cells in a bundle). Each input cells is described by a unique tuple (slotOffset, channelOffset, destination address).

output cells: list of output cells (one or more cells in a bundle). Each output cells is described by a unique tuple (slotOffset, channelOffset, destination address).

### 3.10. Chunk Command

#### 3.10.1. Create.chunk

Create a chunk which consists of one or more unappropriated cells.

To create a chunk, upper layer specifies:

slotframe ID: ID of the slotframe which this chunk belongs to.

ChunkSize: number of the cells which the chunk includes.

SlotBase : the base slotOffset of the chunk.

SlotStep : the incremental of slotOffset in the chunk.

ChannelBase: the base channelOffset of the chunk.

ChannelStep: the incremental of channelOffset in the chunk.

ChunkID is the return value of the command ([I-D.ietf-6tisch-6top-interface]). The chunk is a set of cells in the given slotframe, consisting of (slotOffset(i), channelOffset(i)),  $i=0..ChunkSize-1$ ,  $slotOffset(i) = (slotBase + i * slotStep) \% slotframeLen$ ,  $channelOffset(i) = (channelBase + i * channelStep) \% 16$ . Those cells will be added into ChunkCellList ([I-D.ietf-6tisch-6top-interface]) also.

#### 3.10.2. READ.chunk

Returns the information of a chunk given its ChunkId. The command returns:

slotframe ID: ID of the slotframe which this chunk belongs to.

ChunkSize: number of the cells which the chunk includes.

SlotBase : the base slotOffset of the chunk.

SlotStep : the incremental of slotOffset in the chunk.

ChannelBase: the base channelOffset of the chunk.

ChannelStep: the incremental of channelOffset in the chunk.

Fails if the ChunkId does not exist.

### 3.10.3. Delete.chunk

To delete a chunk, upper layer specifies ChunkID.

It removes the chunk from ChunkList ([I-D.ietf-6tisch-6top-interface]), and also remove those entries corresponding to the cells of the chunk from ChunkCellList([I-D.ietf-6tisch-6top-interface]). In addition, it also causes all of the scheduled cells in the chunk are deleted from CellList ([I-D.ietf-6tisch-6top-interface]) and TSCH schedule as well.

### 3.11. Chunk Cell Command

#### 3.11.1. CREATE.hardcell.fromchunk

Creates one or more hard cells from a chunk. Fails if the cell already exists. A cell is uniquely identified by the tuple (slotframe ID, slotOffset, channelOffset).

To create a hard cell from a chunk which is corresponding to a specific slotframe ID, the upper layer specifies:

chunkID: ID of the chunk which this cell belongs to.

slotOffset: the slotOffset for the cell.

channelOffset: channelOffset for the cell.

LinkOption bitmap: bitmap as defined in [IEEE802154e]

LinkType : as defined in section 6.2.19.3 of [IEEE802154e].

CellType: as defined in Section 2.1

target node address: the address of that node to communicate with over this cell. In case of broadcast cells this is the broadcast address.

TrackID: ID of the track the cell will belong to.

6top schedules the cell and marks it as a hard cell, indicating that it cannot reschedule this cell. In addition, 6top will change the attributes corresponding to the cell in the ChunkCellList, i.e. its CellID is changed to the same CellID in the CellList, and its Status is changed to USED ([I-D.ietf-6tisch-6top-interface]).

The interaction between 6top and MAC layer caused by CREATE.hardcell.fromchunk is same as that caused by CREATE.hardcell (Section 3.1.1).

### 3.11.2. READ.chunkcell

Returns the cell information of a chunk given its ChunkId. For each cell of the chunk, the command returns:

slotOffset: the slotOffset of the cell.

channelOffset: channelOffset of the cell.

cellId: the cellID in the CellList if scheduled.

Status: USED/UNUSED

Fails if the ChunkId does not exist.

### 3.11.3. DELETE.hardcell.fromchunk

To remove a hard cell which comes from a chunk, the upper layer specifies:

slotframe ID: the ID of the slotframe where this cell is installed.

slotOffset: the slotOffset for the cell.

channelOffset: the selected channelOffset for the cell.

LinkOption bitmap: bitmap as defined in [IEEE802154e]

LinkType : as defined in in section 6.2.19.3 of [IEEE802154e].

CellType: as defined in Section 2.1

target node address: the target address of that cell. In case of broadcast cells this is the broadcast address.

TrackID: ID of the track the cell will belong to.

This removes the hard cell from the node's schedule and CellList ([I-D.ietf-6tisch-6top-interface]). In addition, it changes the attributes corresponding to the cell in the ChunkCellList, i.e. its CellID is changed back to FFFF, and its Status is changed to UNUSED ([I-D.ietf-6tisch-6top-interface]).

The interaction between 6top and MAC layer caused by DELETE.hardcell is same as that caused by DELETE.hardcell (Section 3.1.5).

### 3.12. Data Commands

#### 3.12.1. Send.data

The command used by upper layers to queue a packet so underlying TSCH sends it. According to the specific priority, the packet is pushed into a Queue with the equivalent priority or following a criteria out of scope. Once a packet is inserted into a queue it waits to be transmitted by TSCH according to the model defined in Section 2.2. If the queue is full or destination address is not a L2 neighbor of the node, failure to enqueue will be indicated to the caller.

The required parameters are:

src address: L2 address

dest address: L2 unicast or broadcast address

priority: packet priority, usually is consistent with queue priority

message length: the length of the message

message: control message or data message

securityLevel: As defined by [IEEE802154e].

#### 3.12.2. Receive.data

The command is invoked whenever a packet is received and inserted into a reception queue. The method acts as a callback function to notify to the upper layers the received message. Upper layers MUST terminate this indication.

The function has the following parameters:

src address: L2 source address

dest address: L2 unicast or broadcast destination address

priority: packet priority, usually is consistent with queue priority

message length: the length of the message.

message: control message or data message

#### 4. 6top Communication Protocol

This section defines the Information Element (IE) based message formats, and the 6top-to-6top communication time sequences.

##### 4.1. Message Formats

6top has to negotiate the scheduling of soft cells with neighbor nodes. This negotiation happens through 6top-specific TSCH Information Elements, the format of which is defined in this section. For completeness, this section also details the formats of the IEs already defined in [IEEE802154e] and presented here without modification.

6top messages can contain one or more IEs. Section 4.1.1 defines the different IEs used by 6top, both the ones used without modification from [IEEE802154e], and the new ones defined by this document. Section 4.1.2 shows how several IEs are assembled to form the different frames used by 6top.

###### 4.1.1. Information Elements

[IEEE802154e] defines Information elements (IEs). IEs are formatted data objects consisting of an ID, a length, and a data payload used to pass data between layers or devices. [IEEE802154e] defines Header IEs and Payload IEs; 6top only uses Payload IEs. A Payload IE includes one or more IEs, and ends with a termination IE (ID = 0x0f, see [IEEE802154e]).

6top uses the following Information Elements, some defined in [IEEE802154e], others introduced in this document.

Defined in [IEEE802154e] and used by 6top without modification:

TSCH Synchronization IE (Section 4.1.1.1)

TSCH Slotframe and Link IE (Section 4.1.1.2)

TSCH Timeslot Template IE (Section 4.1.1.3)



## TSCH Channel Hopping IE (Section 4.1.1.4)

Defined by 6top:

6top Opcode IE (Section 4.1.1.5)

6top Bandwidth IE (Section 4.1.1.6)

6top TrackID IE (Section 4.1.1.7)

6top Generic Schedule IE (Section 4.1.1.8)

## 4.1.1.1. TSCH Synchronization IE

A Synchronization IE (SyncIE) contains Information allowing a node to synchronize to a TSCH network, including the current ASN and a join priority. Synchronization IE MUST be included in all TSCH Enhanced Beacons.

6top re-uses this IE as defined in [IEEE802154e].

Format of a TSCH Synchronization IE (SyncIE).

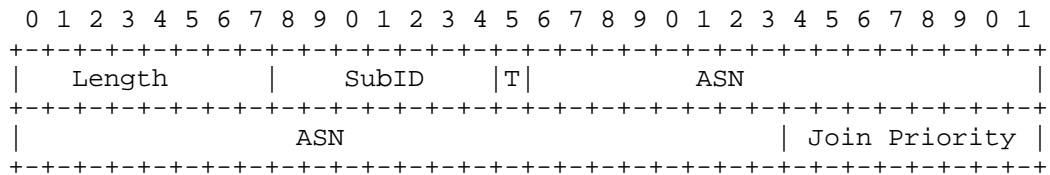


Figure 3

Length=6

SubID=0x1a

T=0, i.e., short type

ASN (5 octets) contains the Absolute Slot Number corresponding to the timeslot in which the TSCH Enhanced Beacon is sent.

The Join Priority can be used by a joining device to select among beaconing devices when multiple beacons are heard. The PAN coordinator's join priority is zero. A lower value of join priority indicates that the device is the preferred one to connect to. As

suggested by [I-D.ietf-6tisch-minimal], the beaconing device's join priority is its DAGRank(rank).

#### 4.1.1.2. TSCH Slotframe and Link IE

The Slotframe and Link IE (FrameAndLinkIE) contains one or more slotframes and their respective cells that a beaconing device advertises to allow other devices to join the network.

6top re-uses this IE as defined in [IEEE802154e].

Format of a TSCH Slotframe and Link IE (FrameAndLinkIE).

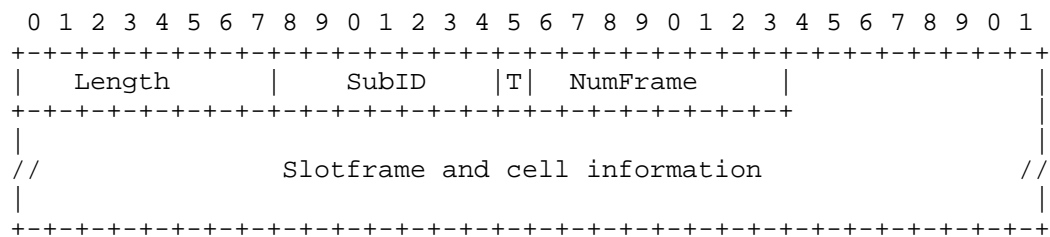


Figure 4

Length=variable

SubID=0x1b

T=0, i.e., short type

NumFrame is set to the total number of slotframe descriptors contained in the TSCH Enhanced Beacon.

Format of a slotframe descriptor.

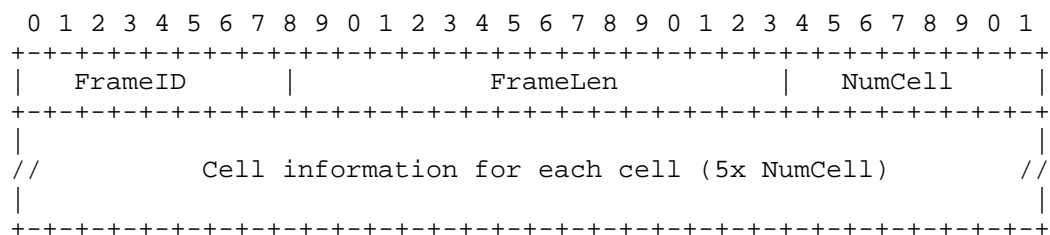


Figure 5

The FrameID field shall be set to the slotframeHandle that uniquely identifies the slotframe.

The FrameLen field shall be set to the size of the slotframe in number of timeslots.

The NumCell field shall be set to the number of cells that belong to the specific slotframe identified by the slotframeHandle.

Format of a Cell information.

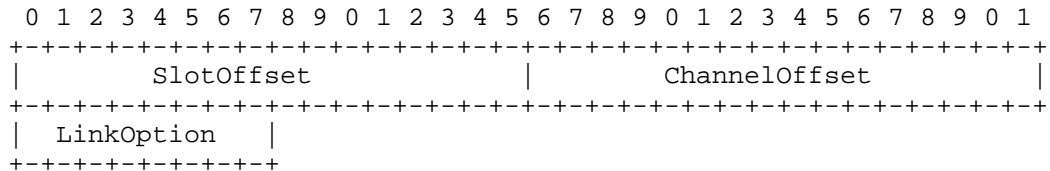


Figure 6

SlotOffset shall be set to the slotOffset of this cell.

ChannelOffset shall be set to the channelOffset of this cell.

LinkOption indicates whether this cell is a TX cell, an RX cell, or a SHARED TX cell, whether the device to which it is being linked is to be used for clock synchronization, and whether this cell is hard cell.

#### 4.1.1.3. TSCH Timeslot Template IE

Timeslot Template IE (SlotTemplateIE) defines Timeslot template being used by the TSCH device.

6top re-uses this IE as defined in [IEEE802154e].

Format of a TSCH Timeslot Template IE (SlotTemplateIE).

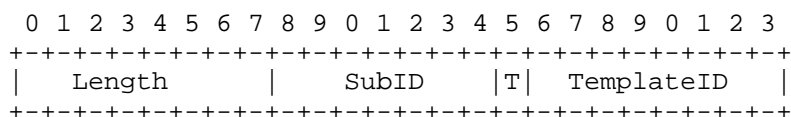


Figure 7

Length=1

SubID=0x1c

T=0, i.e., short type

TemplateID shall be set to a Timeslot template handle. The full timeslot template, which contains the macTimeslotTemplate of TSCH (total 25 octets), MAY be included.(see [IEEE802154e]).

#### 4.1.1.4. TSCH Channel Hopping IE

Channel Hopping IE (ChHoppingIE) defines the Hopping Sequence being used by the TSCH device.

6top re-uses this IE as defined in [IEEE802154e].

### Format of a TSCH Channel Hopping IE (ChHoppingIE).

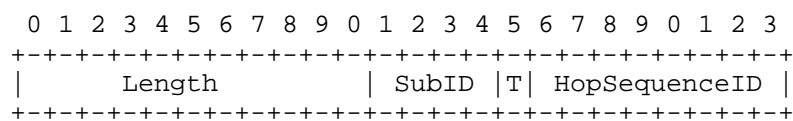


Figure 8

Length=1

SubID=0x09

T=1, i.e., long type

HopSequenceID shall be set to a Hopping Sequence handle. The full Hopping Sequence information MAY be included. (see [IEEE802154e]).

#### 4.1.1.5. 6top Opcode IE

6top Opcode IE (OpcodeIE) defines operation codes of packets in 6top sublayer.

This IE is not present in [IEEE802154e] and is defined by 6top.

Format of a 6top Opcode IE (OpcodeIE).

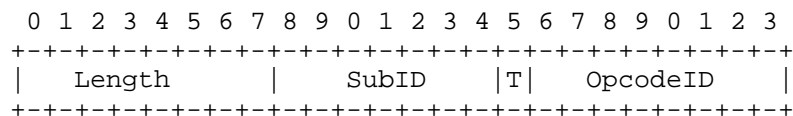


Figure 9

Length=1

SubID=0x41

T=0, i.e., short type

OpcodeID field shall be set to one of the following codes.

0x00: Reserve Soft Cell Request  
 0x01: Reserve Soft Cell Response  
 0x02: Remove Soft Cell Request  
 0x03: Reserve Hard Cell Request  
 0x04: Remove Hard Cell Request

#### 4.1.1.6. 6top Bandwidth IE

Bandwidth IE (BwIE) defines the number of cells to be reserved or actually been reserved.

This IE is not present in [IEEE802154e] and is defined by 6top.

Format of a 6top Bandwidth IE (BwIE).

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1										
Length										SubID										T		FrameID										NumCell									

Figure 10

Length=2

SubID=0x42

T=0, i.e., short type

FrameID MAY be set to the SlotFrameHandle to identify the slotframe from which cells are reserved. FrameID field MAY be set to NOP, which means no specific slotframe is associated.

NumCell shall be set to the number of cells. When BwIE is combined with the OpcodeID of Reserve Soft Cell Request, NumCell presents how many cells are required to reserve; and when BwIE is combined with the OpcodeID of Reserve Soft Cell Response, NumCell presents how many cells are reserved successfully.

## 4.1.1.7. 6top TrackID IE

TrackID IE (TrackIdIE) describes the track which the reserved/removed cell(s) are associated with.

This IE is not present in [IEEE802154e] and is defined by 6top.

Format of a 6top TrackID IE (TrackIdIE).

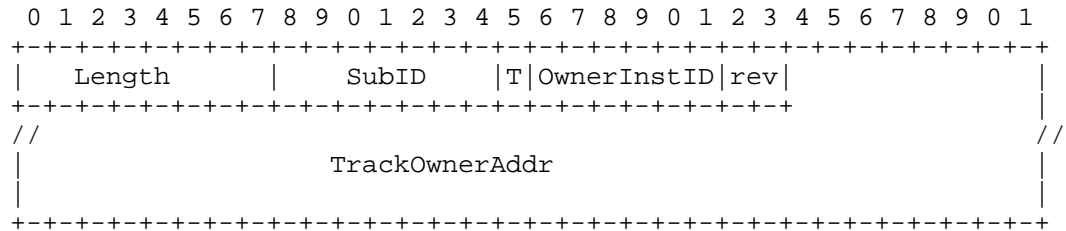


Figure 11

Length=3 or 7. When length=3, TrackOwnerAddr is 2 bytes short address, and when length=7, TrackOwnerAddr is 6 bytes long address.

SubID=0x43

T=0, i.e., short type

The combination of TrackOwnerAddr and OwnerInstId represents a specific TrackID.

## 4.1.1.8. 6top Generic Schedule IE

Generic Schedule IE (ScheduleIE) describes cell sets. In different packets, ScheduleIE represents different information. See Section 4.1.2 for more detail.

This IE is not present in [IEEE802154e] and is defined by 6top.

Format of a 6top Generic Schedule IE (ScheduleIE).

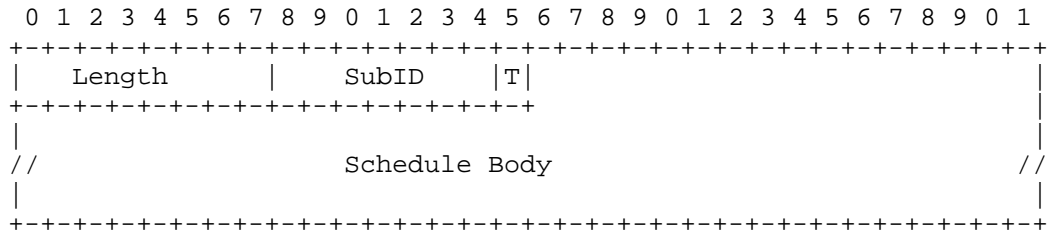


Figure 12

Length=variable

SubID=0x44

T=0, i.e., short type

Schedule Body carries one or more schedule object. An object MAY carry a TLV (Type-Length-Value), which MAY itself comprise other TLVs. TLV format is as follows. Type: 1 byte, Length: 1 byte, Value: variable

The following are some examples of schedule object TLV.

Example 1. Cell Set TLV

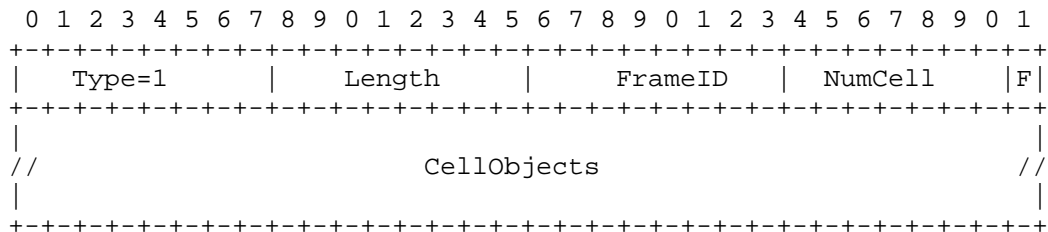


Figure 13

FrameID shall be set to the slotframeHandle that uniquely identifies the slotframe.

NumCell shall be set to the number of cells that belong to the specific slotframe identified by the slotframeHandle.

F=1 means the specified cells equals to what are listed in CellObjects, and F=0 means the specified cells equals to what are not listed in CellObjects.

CellObjects carries the information for one or more cells, including SlotOffset, ChannelOffset, LinkOption (Figure 6).

Example 2. Schedule Matrix TLV

```

    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|  Type=2      |      Length      |      FrameID      |StartSlotOffset|
+-----+-----+-----+-----+-----+-----+-----+-----+
|StartSlotOffset|      NumSlot      |
+-----+-----+-----+-----+-----+-----+
|
//                      SlotBitMap (2x NumSlot)                      //
|
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure 14

FrameID field MUST be set to the slotframeHandle that uniquely identifies the slotframe.

StartSlotOffset field (2 octets) MUST be set to the slotOffset in the specific slotframe identified by the slotframeHandle.

NumSlot field MUST be set to the number of timeslots from StartSlotOffset in the specific slotframe identified by the slotframeHandle.

SlotBitMap (per timeslot) indicates for the given timeslot which channels are specified. For the 16 channels in 2.4GHz band, 2-octets are used to indicate which channel is specified. For example, given a timeslot and a SlotBitmap with value (10001000,00010000); the bitmap represents that ChannelOffset-0, ChannelOffset-4, ChannelOffset-11 are specified.

#### 4.1.2. Packet Formats

This section describes the packets used in 6top to form a network, reserve/maintain bandwidth using soft cells, and reserve/remove hard cells in both the transmitter side and receiver sides. Each of these packets uses one or more IEs defined in Section 4.1.1.

##### 4.1.2.1. TSCH Enhanced Beacon

The TSCH Enhanced Beacon is used to announce the presence of the network and allows new nodes to join. It is an Enhanced Beacon packet defined in [IEEE802154e] with the following Payload IEs:



TSCH Synchronization IE (Section 4.1.1.1)

TSCH Timeslot Template IE (Section 4.1.1.3)

TSCH Channel Hopping IE (Section 4.1.1.4)

TSCH Slotframe and Link IE (Section 4.1.1.2)

Payload IE of TSCH Enhanced Beacon Packet

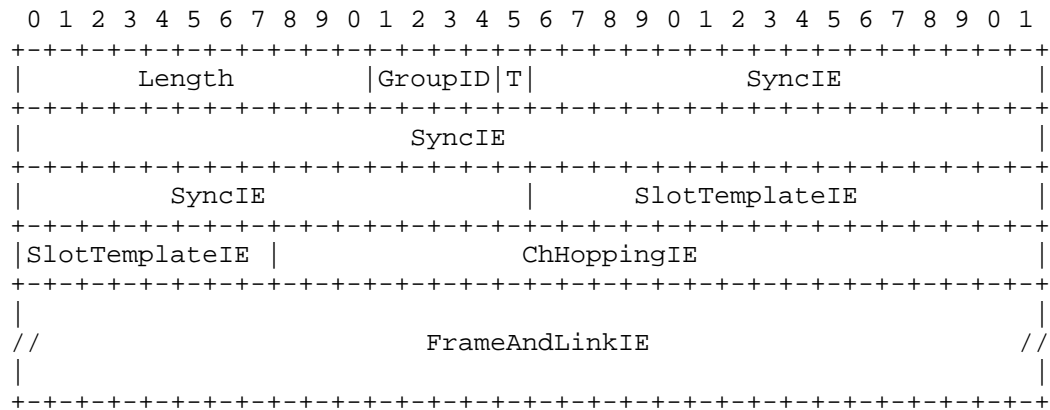


Figure 15

Length=variable

GroupID=0x1, i.e., MLME IE

T=1, i.e., payload IE

See Section 4.1.1.1, Section 4.1.1.3, Section 4.1.1.4,Section 4.1.1.2 for SyncIE, SlotTemplateIE, ChHoppingIE and FrameAndLinkIE.

#### 4.1.2.2. Soft Cell Reservation Request

A Soft Cell Reservation Request packet is a DATA packet defined in [IEEE802154e] with the following payload IE.

## Payload IE of Soft Cell Reservation Request

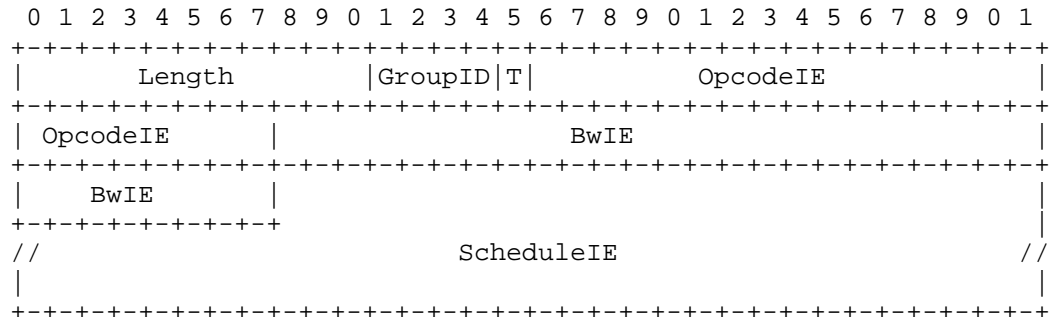


Figure 16

Length=variable

GroupID=0x1, i.e., MLME IE

T=1, i.e., payload IE

The OpcodeID field in the 3-octet OpcodeIE SHOULD be set to 0x00, indicates Reserve Soft Cell Request operation.

The NumCell field in 4-octet BwIE SHOULD be set to the number of cells needed to be reserved.

The ScheduleIE specifies a candidate cell set, from which the cells SHOULD be reserved. ScheduleIE MAY be empty, means there is no constrain on which cells SHOULD not be reserved.

In addition, TrackIdIE can be added in the packet to associate the reserved soft cells to a specific TrackID.

#### 4.1.2.3. Soft Cell Reservation Response

Soft Cell Reservation Response is a DATA packet defined in [IEEE802154e] with the following payload IE.

## Payload IE of Soft Cell Reservation Response

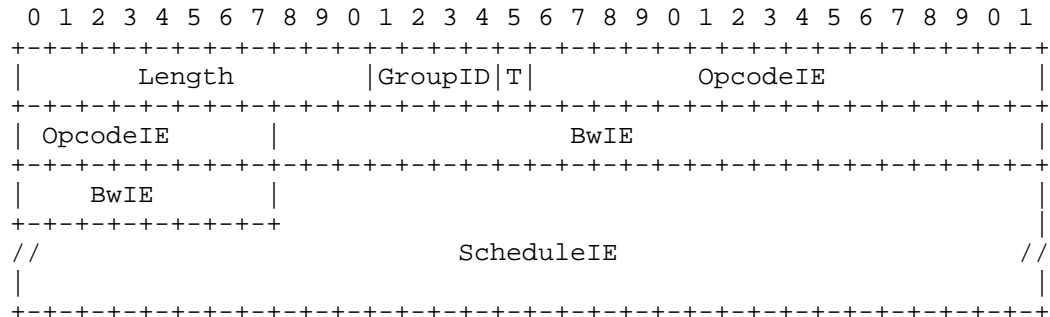


Figure 17

Length=variable

GroupID=0x1, i.e., MLME IE

T=1, i.e., payload IE

The OpcodeID field in the 3-octet OpcodeIE SHOULD be set to 0x01, indicates Reserve Soft Cell Response operation.

The NumCell field in 4-octet BwIE SHOULD be set to the number of cells which have been reserved successfully.

The ScheduleIE SHOULD specify all of the cells which have been reserved successfully.

In addition, TrackIdIE can be added in the packet to associate the reserved soft cells to a specific TrackID.

#### 4.1.2.4. Soft Cell Remove Request

Soft Cell Remove Request is a DATA packet defined in [IEEE802154e] with the following payload IE.

## Payload IE of Soft Cell Remove Request

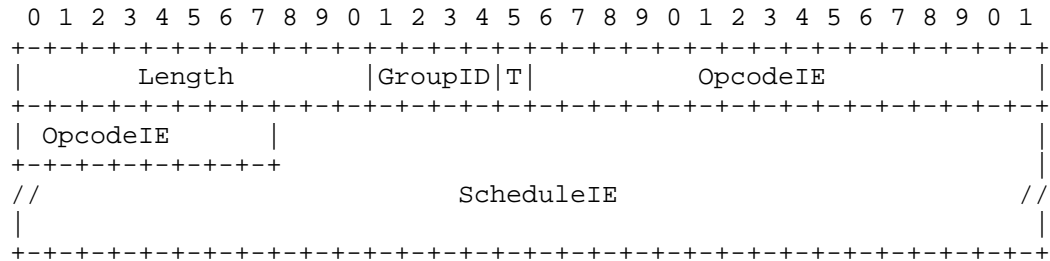


Figure 18

Length=variable

GroupID=0x1, i.e., MLME IE

T=1, i.e., payload IE

The OpcodeID field in the 3-octet OpcodeIE SHOULD be set to 0x02, indicates Remove Soft Cell Request operation.

The ScheduleIE SHOULD specify all the cells that need to be removed.

## 4.1.2.5. Hard Cell Reservation Request

Hard Cell Reservation Request packet is a DATA packet defined in [IEEE802154e] with the following payload IE.

## Payload IE of Hard Cell Reservation Request

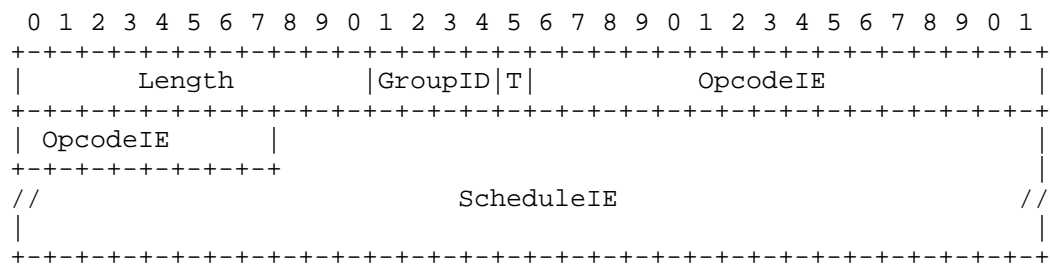


Figure 19

Length=variable

GroupID=0x1, i.e., MLME IE

T=1, i.e., payload IE

The OpcodeID field in the 3-octet OpcodeIE SHOULD be set to 0x03, indicates Reserve Hard Cell Request operation.

The ScheduleIE SHOULD specify all the cell that need to be reserved.

In addition, TrackIdIE can be added in the packet to associate the reserved hard cells to a specific TrackID.

#### 4.1.2.6. Hard Cell Remove Request

Hard Cell Remove Request is a DATA packet defined in [IEEE802154e] with the following payload IE.

Payload IE of Hard Cell Remove Request

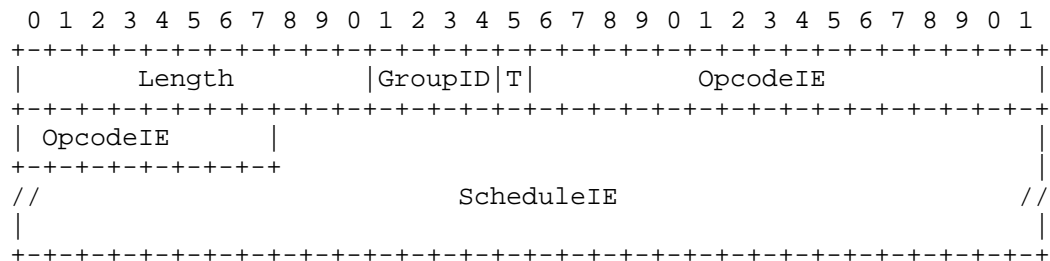


Figure 20

Length=variable

GroupID=0x1, i.e., MLME IE

T=1, i.e., payload IE

The OpcodeID field in the 3-octet OpcodeIE SHOULD be set to 0x04, indicates Remove Hard Cell Request operation.

The ScheduleIE SHOULD specify all the cells that need to be removed.

#### 4.2. Time Sequences

6top neighbors exchange 6top-specific packets in the following cases, each detailed in a subsection.

Network formation (Section 4.2.1)

Creating soft cells (Section 4.2.2)

Deleting soft cells (Section 4.2.3)

Maintaining soft cells (Section 4.2.4)

Creating hard cells (Section 4.2.5)

Deleting hard cells (Section 4.2.6)

#### 4.2.1. Network Formation

Network formation consists of two processes: joining and maintenance.

##### 4.2.1.1. Joining

A node already in the network sends out TSCH Enhanced Beacons periodically.

When a node is joining an existing network, it listens for TSCH Enhanced Beacons. After collecting one or more TSCH Enhanced BEACONS (the format of which is detailed in Section 4.1.2.1), the joining node MUST do the following.

Initialize a neighbor table. Establish a neighbor table and record all of the information described in the TSCH Enhanced BEACONS as its initial schedule with those neighbors.

Select a time source neighbor. According to the Joining Priority described by SyncIEs, the joining node chooses time source neighbors. 6top does not specify the criteria to choose time source neighbors from the Enhanced BEACONS.

Select cells for Enhanced Beacons. The joining node selects one or more cells to indicate in its own Enhanced Beacons, which MAY be the same as the cells used by its neighbors for Enhanced Beacon broadcast, and record those cell(s) into the TSCH schedule with LinkType=ADVERTISING.

Its Enhanced Beacons SHOULD include the cell(s) selected for EB purposes. The EB cells MUST be configured with LinkOption to "Receive" and "Timekeeping", telling its neighbors that the cell is used for broadcast.

Start broadcasting Enhanced Beacon and communicate with neighbors.

#### 4.2.1.2. Maintenance

Nodes MAY broadcast Enhance Beacons on the cells marked with LinkType=ADVERTISING, and listen for Enhanced Beacons from neighbors on the cells with LinkOptions "Receive" and "Timekeeping". If a cell with LinkType=ADVERTISING has both the "Receive" and "Timekeeping" LinkOptions set, which means that the cell is shared by neighbors and itself for broadcasting, then broadcasting Enhanced Beacon has higher priority.

Whenever a node receives an Enhanced Beacon, it SHOULD update its schedule if there is a difference regarding to the cells used for synchronizing with the advertiser of the Enhanced Beacon.

#### 4.2.2. Creating soft cells

The upper layer instructs 6top to schedule one or more soft cells by calling the Create soft cell command. This command can also be called by the monitoring process internal to 6top.

When receiving a Create soft cell command, Node A's 6top sublayer forms a Soft Cell Reservation Request packet which includes the BwIE and ScheduleIE Information Elements. The BwIE indicates the number of cells to be reserved (N1); the ScheduleIE indicates set of a candidate cells from which the new cells SHOULD be selected. If the ScheduleIE is empty, Node A indicates there is no constraint on cell selection.

The Soft Cell Reservation Request is sent to the neighbor (Node B) with whom new cells need to be scheduled. After receiving the Soft Cell Reservation Request, Node B selects the cells from the candidate cell set defined by the ScheduleIE in the Soft Cell Reservation Request, and forms a Soft Cell Reservation Response packet. In the Cell Reservation Response packet, the BwIE indicates the number of cells actually being reserved (N2); the ScheduleIE indicates those reserved cells. If N2 is smaller than N1, node B indicates to node A that there are not enough qualified cells to be reserved. Node B MUST record the reserved cells into its local schedule when sending the Soft Cell Reservation Response. After receiving the Soft Cell Reservation Response, Node A MUST record the reserved cells into its local schedule.

The policy to build a candidate cell set and the policy to select cells from the candidate cell set to reserve are out of scope.

The format of Schedule Body is flexible. For example, Node A can use Cell Set TLV defined in Figure 13 with field 'F' set to '0', and the CellObjects includes all of the cells being used by Node A. In

another word, the cell candidate set is all of the cells not being included in the list defined by CellObjects.

The behavior of the nodes when the soft cells negotiation fails is out of scope.

#### 4.2.3. Deleting soft cells

The upper layer instructs 6top to delete one or more soft cells by calling the Delete soft cell command (Section 3.1.6). This command can also be called by the monitoring process internal to 6top (Section 6).

When receiving a Delete soft cell command, Node A's 6top sublayer selects cells to be removed from its local schedule, and creates a Soft Cell Remove Request, which includes a ScheduleIE Information Element. The ScheduleIE indicates which specific cells to remove with a neighbor (Node B). The cells specified in the ScheduleIE SHOULD be removed from local schedule of Node A when the Soft Cell Remove Request is sent to Node B. When receiving the Soft Cell Remove Request, the cells specified in the ScheduleIE SHOULD be removed from the local schedule of Node B.

The policy to select cells corresponding to a Delete soft cell command is out of scope.

#### 4.2.4. Maintaining soft cells

The monitoring process internal to 6top (Section 6) is responsible for monitoring and re-scheduling soft cells to meet some QoS requirements. The monitoring process MAY issue a soft cell Maintenance command, which indicate a set of cells to be re-allocated in the TSCH schedule.

When receiving a soft cell Maintenance command, 6top initializes a Soft Cell Remove Request (Section 4.2.3) with the neighbor in question, followed by a Soft Cell Reservation Request (Section 4.2.2).

#### 4.2.5. Creating hard cells

The upper layer instructs 6top to create one or more hard cells by calling the Create hard cell command.

When receiving a Create hard cell command, Node A's 6top sublayer creates a Hard Cell Reservation Request, including a ScheduleIE. The ScheduleIE indicates which specific cells with a neighbor (Node B) to be added. The cells specified in the ScheduleIE SHOULD be added in



local schedule of Node A while the Hard Cell Reserve Request is sent to Node B. When receiving the Hard Cell Reserve Request, the cells specified in the ScheduleIE SHOULD be added in the local schedule of Node B.

#### 4.2.6. Deleting hard cells

The upper layer instructs 6top to delete one or more hard cells by calling the Delete hard cell command.

When receiving a Delete hard cell command, Node A's 6top sublayer creates a Hard Cell Remove Request, including a ScheduleIE. The ScheduleIE indicates which specific cells with a neighbor (Node B) to be removed. The cells specified in the ScheduleIE SHOULD be removed from local schedule of Node A while the Hard Cell Remove Request is sent to Node B. When receiving the Hard Cell Remove Request, the cells specified in the ScheduleIE SHOULD be removed from the local schedule of Node B.

### 5. Statistics

The 6top Statistics Function (SF) is responsible for collecting statistics, which it can provide to an upper layer and the Monitoring Function (Section 6).

#### 5.1. Statistics Metrics

6top is in charge of keeping statistics from a set of metrics gathered from the behavior of the TSCH layer.

The statistics data related to node states and cell metrics SHOULD be provided to upper layer for management, e.g., for RPL to calculate the node's Rank or for GMPLS to the required bandwidth is met. The specific algorithm to generate the statistics is out of scope. However, the statistics component SHOULD include the following metrics:

1. LinkThroughput: associated with a link, Node A->Node B. For example, LinkThroughput can be calculated with:  
$$\text{SUM}(\text{NumOfCell}(i) * \text{NumOfBytePerPacket}) / (\text{FrameLen}(i) * \text{SlotDuration})$$
where NumOfCell(i) is the total number of cells from Node A to Node B in Slotframe-i, FrameLen(i) is the length of Slotframe-i. The unit is Byte/second.
2. Latency: associated with a link, Node A->Node B. For example, latency can be expressed as Minimum and Maximum Latency. Minimum Latency = Min(MinNumOfSlot(i), i=1..) \* SlotDuration and Maximum Latency = Max(MaxNumOfSlot(i), i=1..) \* SlotDuration where,

MinNumOfSlot(i) and MaxNumOfSlot(i) are the minimum or maximum number of timeslots between two dedicated cells from Node A to Node B in Slotframe-i, respectively.

3. LinkQuality. For example, average LQI, ETX, PDR, RSSI.
4. TrafficLoad. For example, Queue Full Rate, Queue Empty Rate.
5. NodeEnergy. For example,  $E_E = E_{bat} / [E_0 (T-t)/T]$ .

## 5.2. Statistics Configuration

The Statistics Function SHOULD be configurable. The configuration parameters SHOULD include:

LinkQualityStatisticsEn

TafficLoadStatisticsEn

DeviceStatisticsEn

6top statistics function is enabled/disabled and configured by the commands defined in Section 3.4

## 6. Monitoring

The 6top Monitoring Function (MF) is responsible for monitoring cell quality, traffic load, and issuing soft cell Maintenance commands, or Create/Delete soft cell commands. The data provided by the Statistics Function MAY be used as an input of MF in taking a monitoring decision.

### 6.1. Monitor Configuration

Monitoring Function SHOULD be configurable. The configuration parameters SHOULD include:

MaintainCellEn.

CreateDeleteCellEn.

QosLevel. QosLevel SHOULD associate with specific neighbor address. QosLevel MAY reflect the latency constraint, cell quality constraint, and so on. The value of QosLevel works as the bandwidth redundancy coefficient.

The 6top monitoring function is enabled/disabled and configured by the commands defined in Section 3.3

## 6.2. Actuation

The cell quality statistics MAY be used to generate soft a cell Maintenance command, which triggers a soft cell Maintenance procedure (see Section 4.2.4). The traffic load statistics MAY be used to generate internal Create (resp. Delete) soft cell commands, which triggers a soft cell Reservation (resp. Remove) process (see Section 4.2.2 and Section 4.2.3).

The policy to generate the soft cell Maintenance command and the policy to generate Create/Delete soft cell commands is out of scope.

The policy to generate Create/Delete soft cell commands MAY take QoSLevel into account. For example, there are two slotframes existing, Slotframe-1 consists of 32 timeslots, Slotframe-2 consists of 96 timeslots; timeslot duration is 10ms; QoSLevel=1.5. If, from the traffic load statistics, MF determines that 2 packet/second SHOULD be added, then the MF generates a Create soft cell command, where FrameID=2, NumCell=3.

## 7. References

### 7.1. Normative References

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

### 7.2. Informative References

[I-D.ietf-6tisch-tsch]  
Watteyne, T., Palattella, M., and L. Grieco, "Using IEEE802.15.4e TSCH in an LLN context: Overview, Problem Statement and Goals", draft-ietf-6tisch-tsch-00 (work in progress), November 2013.

[I-D.ietf-6tisch-architecture]  
Thubert, P., Watteyne, T., and R. Assimiti, "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-architecture-02 (work in progress), June 2014.

[I-D.ietf-6tisch-terminology]  
Palattella, M., Thubert, P., Watteyne, T., and Q. Wang, "Terminology in IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-terminology-01 (work in progress), February 2014.

[I-D.ietf-6tisch-minimal]

Vilajosana, X. and K. Pister, "Minimal 6TiSCH Configuration", draft-ietf-6tisch-minimal-01 (work in progress), June 2014.

[I-D.ietf-6tisch-6top-interface]

Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top) Interface", draft-ietf-6tisch-6top-interface-00 (work in progress), March 2014.

[I-D.wang-6tisch-6top-sublayer]

Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top)", draft-wang-6tisch-6top-sublayer-00 (work in progress), February 2014.

[I-D.ietf-6tisch-coap]

Sudhaakar, R. and P. Zand, "6TiSCH Resource Management and Interaction using CoAP", draft-ietf-6tisch-coap-00 (work in progress), May 2014.

### 7.3. 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.

[OpenWSN]

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 , August 2012.

[label-switching-154e]

Morell, A., Vilajosana, X., Lopez-Vicario, J., and T. Watteyne, "Label Switching over IEEE802.15.4e Networks. Transactions on Emerging Telecommunications Technologies", June 2013.

## Authors' Addresses

Qin Wang (editor)  
Univ. of Sci. and Tech. Beijing  
30 Xueyuan Road  
Beijing, Hebei 100083  
China

Phone: +86 (10) 6233 4781  
Email: wangqin@ies.ustb.edu.cn

Xavier Vilajosana  
Universitat Oberta de Catalunya  
156 Rambla Poblenou  
Barcelona, Catalonia 08018  
Spain

Phone: +34 (646) 633 681  
Email: xvilajosana@uoc.edu

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

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