6TiSCH
Internet-Draft
Intended status: Informational
Expires: September 8, 2015

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

March 7, 2015

6TiSCH On-the-Fly Scheduling
draft-dujovne-6tisch-on-the-fly-05

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 soft cells between neighbor nodes, based on the specific application constraints to be satisfied. When using OTF, softcell 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

Table of Contents

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 algorithm 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 soft cells. It

also defines two threshold values for bounding the number of triggered 6top allocate/delete commands. 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 soft cells) 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, OTF 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.

OTF is a Layer-3 Mechanism, and as such, it operates on L3 links, on the best effort track, i.e. with TrackID=00, as defined in [I-D.wang-6tisch-6top]. Inside an intermediate node, a track is uniquely associated to a pair of bundles: one incoming bundle, and one outgoing bundle. For an IP link, the two bundle are identified by the same peer mac addresses. For instance (macA, macB, TrackID=00) and (macB, macA, TrackID=00) will be the two bundles associated to the L3 link between node A and node B. The cells on the best effort track can be used for forwarding any packet in the queue, regardless of the specific L2 bundle (and thus, end-to-end L2 track) the packet 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.

This document introduces the following parameters for describing the behavior of the OTF allocation policy:

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

REQUIREDCELLS: Number of cells requested by OTF to 6top, a non-negative value. How this is computed is out of the scope. It MAY be an instantaneous request, or a value averaged on several measurements.

OTFTHRESHLOW: Threshold parameter introducing cell over-provisioning in the allocation policy. It is a non-negative value expressed as number of cells. Which value to use is application-specific and out of scope.

OTFTHRESHHIGH: Threshold parameter introducing cell under-provisioning in the allocation policy. It is a non-negative value expressed as number of cells. Which value to use is application-specific and out of scope.

The OTF allocation policy compares the number of required cells against the number of scheduled ones, using the OTF threshold for bounding the signaling overhead due to negotiations of new cells. In details:

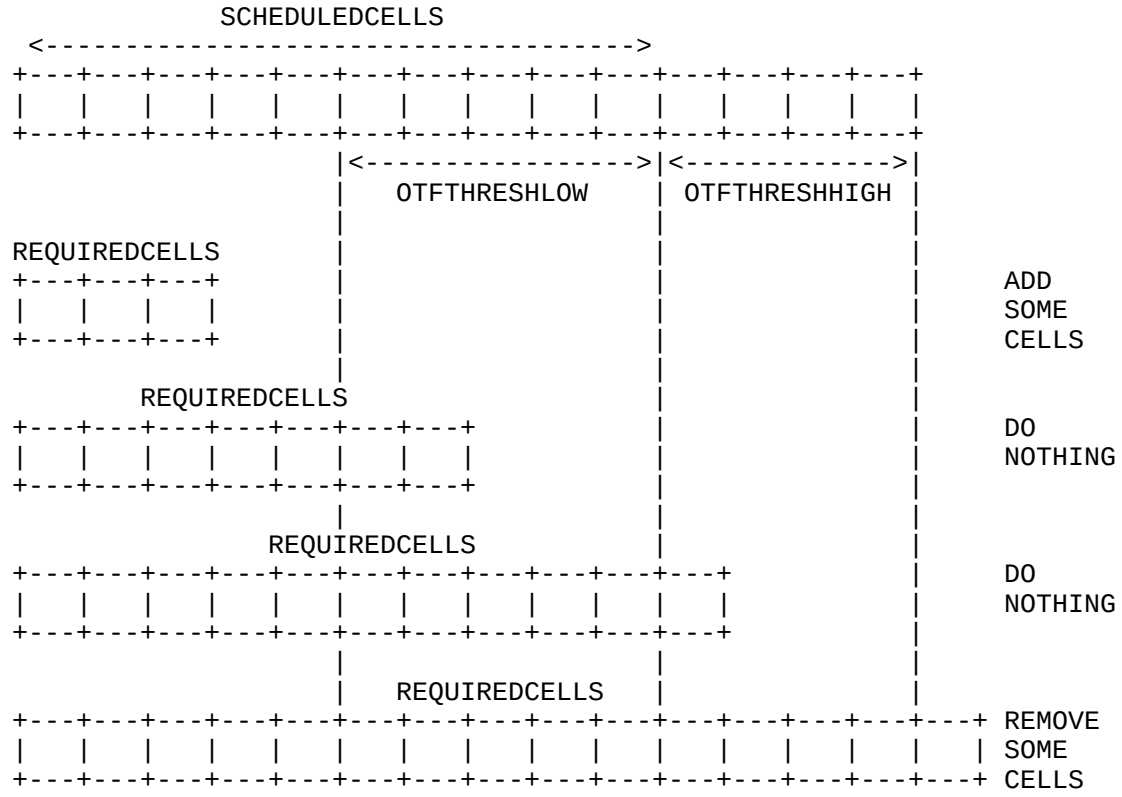


Figure 1: Relation among the OTF parameters used for triggering add/ remove 6top commands

1. If REQUIREDCELLS is greater than (SCHEDULEDCELLS + OTFTHRESHHIGH), OTF asks 6top to add one or more soft cells to the bundle on the best effort track.

2. If REQUIREDCELLS is greater or equal than (SCHEDULEDCELLS - OTFTHRESHLOW), and it is lower than or equal to (SCHEDULEDCELLS + OTFTHRESHHIGH), OTF does not perform any bundle resizing, since the scheduled cells are sufficient for managing the current traffic conditions.

3. If REQUIREDCELLS is lower than (SCHEDULEDCELLS - OTFTHRESHLOW), OTF asks 6top to delete one or more soft cells from the bundle on the best-effort track.

When both OTFTHRESHLOW and OTFTHRESHHIGH equal 0, any discrepancy between REQUIREDCELLS and SCHEDULEDCELLS triggers a 6top negotiation

of soft cells. Other values for the thresholds values reduce the number of triggered 6top negotiations.

The number of soft cells to be scheduled/deleted for bundle resizing is out of the scope of this document and implementation-dependant.

3. Allocation method

Beyond the allocation policies that describe the approach used by OTF for fulfilling the node bandwidth requests, the OTF framework also includes the Allocation Method that specify how OTF issues commands to the 6top sublayer. As specified in [I-D.wang-6tisch-6top], 6top provides a set of commands that allows OTF to allocate/delete soft cells. Such commands are used by the OTF soft cell allocation method.

With the soft cell allocation method, OTF can ask 6top to reserve one (or N > 1) soft cell(s) on the best effort L3 boundle, between two neighbor nodes. The 6top layer allocates and maintains these cells. If a L3 bundle with TrackID=00 was already reserved between the same pair of neighbors, 6top translates the OTF request into a bundle resize request. The newly allocated cell increases 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 (or N > 1) from the best effort track, i.e. to decrease the size of the best effort L3 bundle. If no bundle with TrackID=00 exists when 6top receives the OTF request, then the 6top softcell create command generates a new bundle of size 1.

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 specifies the track the cell belongs to (i.e., best effort track, TrackID=00), but not its slotOffset nor the channelOffset. If at least one cell on the best effort L3 bundle 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. By using the CREATE.softcell command, OTF can ask 6top to add multiple softcells on the best effort L3 bundle. 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 L3 bundle size of the already existing best-effort

bundle. By using the DELETE.softcell command, OTF can ask 6top to delete cells from the 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 structures 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 drops 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, e.g. based on the queue length increase, OTF may ask 6top to add some cells, in order to increase the available bandwidth.

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 (DELETE.softcell) 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 soft cell allocation. Both type of events, the one which triggers the algorithm and the ones which are generated by the execution of the algorithm are called OTF events.

A set of parameters P(E): parameters used to define E and its triggering conditions;

a set of triggering variables V(E): variables that can trigger the event;

a set of triggering conditions C(E): conditions to satisfy on the variables V(E) to trigger E;

a set of process handlers H(E): handlers required to respond and process the triggering conditions C(E).

To illustrate how P(E), V(E), C(E) and H(E) can be used to define a real event, the allocation policy described in Sec. 2 is considered hereby.

P(E) consists of the OTFTHRESHLOW and OTFTHRESHHIGH parameters (P1 and P2, respectively);

V(E) consists of the REQUIREDCELLS and SCHEDULEDCELLS parameters (V1 and V2, respectively);

C(E) consists of the following conditions:

C1: V1 > V2+P2

C2: V1 <=V2-P1

H(E) consists of the following handlers (one handler for each triggering condition)

H1(C1): OTF asks 6top to add one or more soft cells to the L3 best effort bundle.

H2(C2): OTF asks 6top to delete one or more soft cells from the L3 best effort bundle.

7. Bandwidth Estimation Algorithms

OTF supports different bandwidth estimation algorithms that can be used by a node in a 6TiSCH network for checking the statistics provided by 6top and the actual bandwidth usage. By doing so, one can adapt (increase or decrease) the number of scheduled soft cells for a given pair of neighbors (e.g., parent node and its child), according to their specific requirements. 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.

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 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, i.e., it uses OTFTHRESHLOW = 0 and OTFTHRESHHIGH = 0.

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.

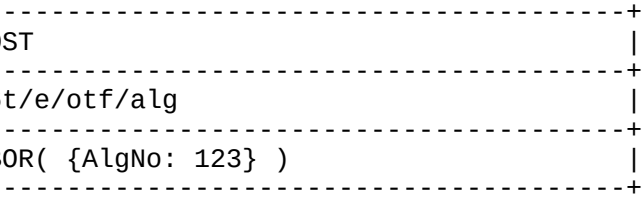


Figure 2: Algorithm number POST message

To obtain the current algorithm number:

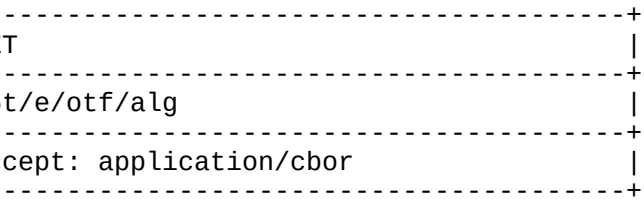


Figure 3: Algorithm number GET message

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

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

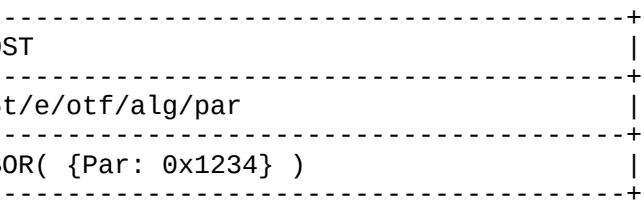


Figure 4: Algorithm number POST message

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-03 (work in progress), January 2015.

[I-D.ietf-6tisch-architecture]
Thubert, P., Watteyne, T., Struik, R., and M. Richardson, "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-architecture-05 (work in progress), January 2015.

[I-D.ietf-6tisch-tsch]
Watteyne, T., Palattella, M., and L. Grieco, "Using IEEE802.15.4e TSCH in an IoT context: Overview, Problem Statement and Goals", draft-ietf-6tisch-tsch-05 (work in progress), January 2015.

[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) Amendament 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

DetNet
Internet-Draft
Intended status: Standards Track
Expires: September 10, 2015

N. Finn
P. Thubert
Cisco

March 9, 2015

Deterministic Networking Architecture
draft-finn-detnet-architecture-00

Abstract

Deterministic Networking (DetNet) provides a capability to carry specified unicast or multicast data streams for real-time applications with extremely low data loss rates and maximum latency. Techniques used include: 1) reserving data plane resources for individual (or aggregated) DetNet streams in some or all of the relay systems (bridges or routers) along the path of the stream; 2) providing fixed paths for DetNet streams that do not rapidly change with the network topology; and 3) sequentializing, replicating, and eliminating duplicate packets at various points to ensure the availability of at least one path. The capabilities can be managed by configuration, or by manual or automatic network management.

Table of Contents

1. Introduction

Operational Technology (OT) refers to industrial networks that are typically used for monitoring systems and supporting control loops, as well as movement detection systems for use in process control (i.e., process manufacturing) and factory automation (i.e., discrete manufacturing). Due to its different goals, OT has evolved in

parallel but in a manner that is radically different from IT/ICT, focusing on highly secure, reliable and deterministic networks, with limited scalability over a bounded area.

The convergence of IT and OT technologies, also called the Industrial Internet, represents a major evolution for both sides. The work has already started; in particular, the industrial automation space has been developing a number of Ethernet-based replacements for existing digital control systems, often not packet-based (fieldbus technologies).

These replacements are meant to provide similar behavior as the incumbent protocols, and their common focus is to transport a fully characterized flow over a well-controlled environment (i.e., a factory floor), with a bounded latency, extraordinarily low frame loss, and a very narrow jitter. Examples of such protocols include PROFINET, ODVA Ethernet/IP, and EtherCAT.

In parallel, the need for determinism in professional and home audio/video markets drove the formation of the Audio/Video Bridging (AVB) standards effort of IEEE 802.1. With the explosion of demand for connectivity and multimedia in transportation in general, the Ethernet AVB technology has become one of the hottest topics, in particular in the automotive connectivity. It is finding application in all elements of the vehicle from head units, to rear seat entertainment modules, to amplifiers and camera modules. While aimed at less critical applications than some industrial networks, AVB networks share the requirement for extremely low packet loss rates and ensured finite latency and jitter.

Other instances of in-vehicle deterministic networks have arisen as well for control networks in cars, trains and buses, as well as avionics, with, for instance, the mission-critical "Avionics Full-Duplex Switched Ethernet" (AFDX) that was designed as part of the ARINC 664 standards. Existing automotive control networks such as the LIN, CAN and FlexRay standards were not designed to cover these increasing demands in terms of bandwidth and scalability that we see with various kinds of Driver Assistance Systems (DAS) and new multiplexing technologies based on Ethernet are now getting traction.

The generalization of the needs for more deterministic networks have led to the IEEE 802.1 AVB Task Group becoming the Time-Sensitive Networking (TSN) Task Group (TG), with a much-expanded constituency from the industrial and vehicular markets. Along with this expansion, the networks in consideration are becoming larger and structured, requiring deterministic forwarding beyond the LAN boundaries. For instance, Industrial Automation segregates the network along the broad lines of the Purdue Enterprise Reference

Architecture (PERA), using different technologies at each level, and public infrastructures such as Electricity Automation require deterministic properties over the Wide Area. The realization is now coming that the convergence of IT and OT networks requires Layer-3, as well as Layer-2, capabilities.

The present architecture is the result of a collaboration of the IETF and the IEEE and implements an abstract model that can be applicable both at Layer-2 and Layer-3, and along segments of different technologie. With this new work, a path may span, for instance, across a (limited) number of 802.1 bridges and then a (limited) number of IP routers. In that example, the IEEE 802.1 bridges may be operating at Layer-2 over Ethernet whereas the IP routers may be 6TiSCH nodes operating at Layer-2 and/or Layer-3 over the IEEE 802.15.4e MAC.

Many applications of interest to Deterministic Networking require the ability to synchronize the clocks in end systems to a sub-microsecond accuracy. Some of the queue control techniques defined in Section 4.7 also require time synchronization among relay systems. The means used to achieve time synchronization are not addressed in this document.

2. Terminology

The follwing special terms are used in this document in order to avoid the assumption that a given element in the archetecture does or does not have Internet Protocol stack, functions as a router or a bridge, or otherwise plays a particular role at Layer-3 or higher:

bridge
: A Customer Bridge as defined by IEEE 802.1Q [IEEE802.1Q-2011].

circuit
: A trail of configuration from talker to listener(s) through relay systems associated with a DetNet stream, required to deliver the benefits of DetNet.

end system
: Commonly called a "host" in IETF documents, and an "end station" is IEEE 802 documents. End systems of interest to this document are talkers and listeners.

listener
: An end system capable of sinking a DetNet stream.

relay system

A router or a bridge.

stream
: A DetNet stream is a sequence of packets from a single talker, through some number of relay systems to one or more listeners, that is limited by the talker in its maximum packet size and transmission rate, and can thus be ensured the DetNet Quality of Service (QoS) from the network.

talker
: An end system capable of sourcing a DetNet stream.

3. Providing the DetNet Quality of Service

DetNet Quality of Service is expressed in terms of:

- Minimum and maximum end-to-end latency from talker to listener;
- Probability of loss of a packet, assuming the normal operation of the relay systems and links;
- Probability of loss of a packet in the event of the failure of a relay system or link.

It is a distinction of DetNet that it is concerned solely with worst-case values for all of the above parameters. Average, mean, or typical values are of no interest, because they do not affect the ability of a real-time system to perform its tasks.

Three techniques are employed by DetNet to achieve these QoS parameters:

a. Zero congestion loss (Section 3.1). Network resources such as link bandwidth, buffers, queues, shapers, and scheduled input/output slots are assigned in each relay system to the use of a specific DetNet stream or group of streams. Note that, given a finite amount of buffer space), zero congestion loss necessarily ensures a maximum end-to-end latency. Depending on the method employed, a minimum latency can also be achieved.

b. Pinned-down paths (Section 3.2). Point-to-point paths or point-to-multipoint trees through the network from a talker to one or more listeners can be established, and DetNet streams assigned to follow a particular path or tree.

c. Packet replication and deletion (Section 3.3). End systems and/or relay systems can sequence number, replicate, and eliminate replicated packets at multiple points in the network in order to

ensure that one (or more) equipment failure events still leave at least one path intact for a DetNet stream.

These three techniques can be applied independently, giving eight possible combinations, including none (no DetNet), although some combinations are of wider utility than others. This separation keeps the protocol stack coherent and maximizes interoperability with existing and developing standards in this (IETF) and other Standards Development Organizations. Some examples of typical expected combinations:

- Pinned-down paths (a) plus packet replication (b) are exactly the techniques employed by [HSR-PRP]. Pinned-down paths are achieve by limiting the physical topology of the network, and the sequentialization, replication, and duplicate elimination facilitated by packet tags added at the front or the end of Ethernet frames.

- Zero congestion loss (a) alone is is offered by IEEE 802.1 Audio Video bridging [IEEE802.1BA-2011]. As long as the network suffers no failures, near-zero (at best, zero) congestion loss can be achieved through the use of a reservation protocol (MSRP) and shapers in every relay system (bridge).

- Using all three together gives maximum protection.

There are, of course, simpler methods available (and employed, today) to achieve levels of latency and packet loss that are satisfactory for many applications. However, these methods generally work best in the absence of any significant amount of non-critical traffic in the network (if, indeed, such traffic is supported at all), or work only if the critical traffic constitutes only a small portion of the network’s theoretical capacity, or work only if all systems are functioning properly, or in the absence of actions by end systems that disrupt the network’s operations.

There are any number of methods in use, defined, or in progress for accomplishing each of the above techniques. It is expected that this DetNet Architecture will assist various vendors, users, and/or "vertical" Standards Development Organizations (dedicated to a single industry) to make selections among the available means of implementing DetNet networks.

3.1. Zero Congestion Loss

The primary means by which DetNet achieves its QoS assurances is to completely eliminate congestion at an output port as a cause of packet loss. Given that a DetNet stream cannot be throttled, this

can be achieved only by the provision of sufficient buffer storage at each hop through the network to ensure that no packets are dropped due to a lack of buffer storage.

Ensuring adequate buffering requires, in turn, that the talker, and every relay system along the path to the listener (or nearly every relay system -- see Section 4.5.2) be careful to regulate its output to not exceed the data rate for any stream, except for brief perios when making up for interfering traffic. Any packet sent ahead of its time potentially adds to the number of buffers required by the next hop, and may thus exceed the resources allocated for a particular stream.

The low-level mechanisms described in Section 4.7 provide the necessary regulation of transmissions by an edge system or relay system to ensure zero congestion loss. Of course, the reservation of the bandwidth and buffers for a stream requires the provisioning described in Section 4.12.

3.2. Pinned-down paths

In networks controlled by typical peer-to-peer protocols such as IEEE 802.1 ISIS bridged networks or ETF OSPF routed networks, a network topology event in one part of the network can impact, at least briefly, the delivery of data in parts of the network remote from the failure or recovery event. Thus, even redundant paths through a network, if controlled by the typical peer-to-peer protocols, do not eliminate the chances of brief losses of contact. For this reason, many real-time networks rely on physical rings of two-port devices, with a relatively simple ring control protocol. This both minimizes recovery time and easily supports redundant paths. Of course, this comes at the cost of increased hop count, and thus latency, for the typical path.

In order to get the advantages of low hop count and still ensure against even brief losses of connectivity, DetNet employs pinned-down paths, where the path taken by a given DetNet stream does not change, at least immediately, and likely not at all, in response to network topology events. When combined with seamless redundancy (Section 3.3), this results in a high likelihood of continuous connectivity.

3.3. Seamless Redundancy

After congestion loss has been eliminated, the most important causes of packet loss are random media and/or memory faults and equipment failures.

Seamless redundancy involves three capabilities:

- Adding sequence numbers to the packets of a DetNet stream.
- Replicating these packets and, typically, sending them along at least two different paths to the listener(s).
- Discarding duplicated packets.

In the simplest case, this amounts to replicating each packet in a talker that has two interfaces, and conveying them through the network, along separate paths, to the similarly dual-homed listeners, that discard the extras. This ensures that one path (with zero congestion loss) remains, even if some relay system fails.

Alternatively, relay systems in the network can provide replication and elimination facilities at various points in the network, so that multiple failures can be accommodated.

This is shown in the following figure, where the two relay systems each replicate (R) the DetNet stream on input, sending the stream to both the other relay system and to the end system, and eliminated duplicates (E) on the output interface to the right-hand end system. Any one links in the network can fail, and the Detnet stream can still get through. Furthermore, two links can fail, as long as they are in different segments of the network.

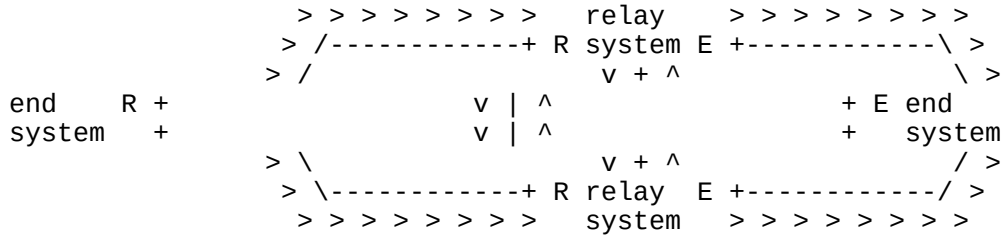


Figure 1

4. DetNet Architecture

Traffic Engineering Architecture and Signaling (TEAS) [TEAS] defines traffic-engineering architectures for generic applicability across packet and non-packet networks. From TEAS perspective, Traffic Engineering (TE) refers to techniques that enable operators to control how specific traffic flows are treated within their networks.

Because if its very nature of establishing pinned-down optimized paths, Deterministic Networking can be seen as a new, specialized

branch of Traffic Engineering, and inherits its architecture with a separation into planes.

The Deterministic Networking architecture is thus composed of three planes, a (User) Application Plane, a Controller Plane, and a Network Plane, which echoes that of Software-Defined Networking (SDN): Layers and Architecture Terminology [RFC7426] which is represented below:

SDN Layers and Architecture Terminology per RFC 7426

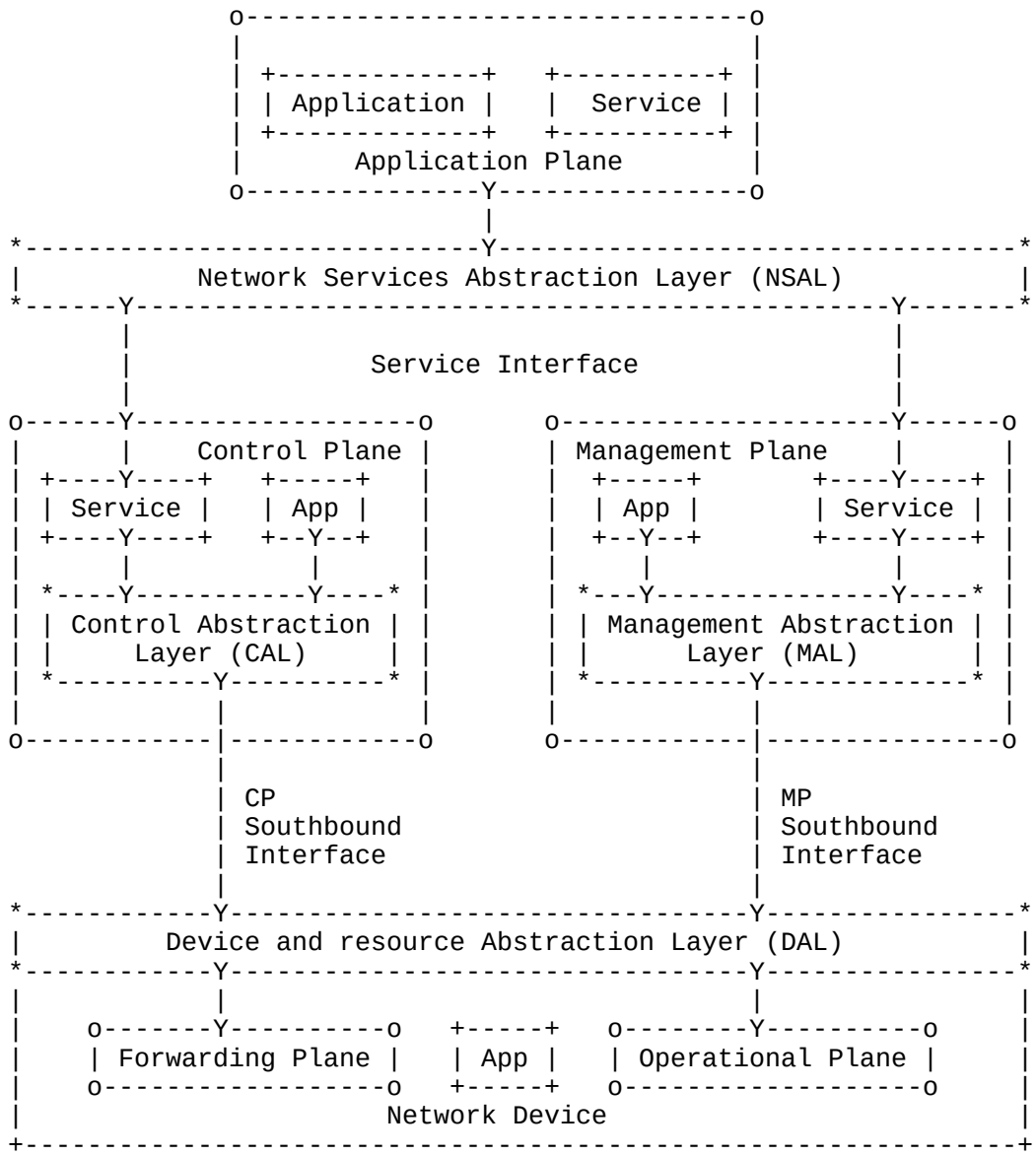


Figure 2

4.1. The Application Plane

Per [RFC7426], the Application Plane includes both applications and services. In particular, the Application Plane incorporates the User Agent, a specialized application that interacts with the end user / operator and performs requests for Deterministic Networking services via an abstract Stream Management Entity, (SME) which may or may not be collocated with (one of) the end systems.

At the Application Plane, a management interface enables the negotiation of streams between end systems. An abstraction of the stream called a Traffic Specification (TSpec) provides the representation. This abstraction is used to place a reservation over the (Northbound) Service Interface and within the Application plane. It is associated with an abstraction of location, such as IP addresses and DNS names, to identify the end systems and eventually specify intermediate relay systems.

4.2. The Controller Plane

The Controller Plane corresponds to the aggregation of the Control and Management Planes in [RFC7426], though Common Control and Measurement Plane (CCAMP) [CCAMP] makes an additional distinction between management and measurement. When the logical separation of the Control, Measurement and other Management entities is not relevant, the term Controller Plane is used for simplicity to represent them all, and the term controller refers to any device operating in that plane, whether is it a Path Computation entity or a Network Management entity (NME). The Path Computation Element (PCE) [PCE] is a core element of a controller, in charge of computing Deterministic paths to be applied in the Network Plane.

A (Northbound) Service Interface enables applications in the Application Plane to communicate with the entities in the Controller Plane.

One or more PCE(s) collaborate to implement the requests from the SME as Per-Stream Per-Hop Behaviors installed in the relay systems for each individual streams. The PCEs place each stream along a deterministic sequence of relay systems so as to respect per-stream constraints such as security and latency, and optimize the overall result for metrics such as an abstract aggregated cost. The deterministic sequence can typically be more complex than a direct sequence and include redundancy path, with one or more packet replication and elimination points.

4.3. The Network Plane

The Network Plane represents the network devices and protocols as a whole, regardless of the Layer at which the network devices operate.

The network Plane comprises the Network Interface Cards (NIC) in the end systems, which are typically IP hosts, and relay systems, which are typically IP routers and switches. Network-to-Network Interfaces such as used for Traffic Engineering path reservation in [RFC3209], as well as User-to-Network Interfaces (UNI) such as provided by the Local Management Interface (LMI) between network and end systems, are all part of the Network Plane.

A Southbound (Network) Interface enables the entities in the Controller Plane to communicate with devices in the Network Plane. This interface leverages and extends TEAS to describe the physical topology and resources in the Network Plane.

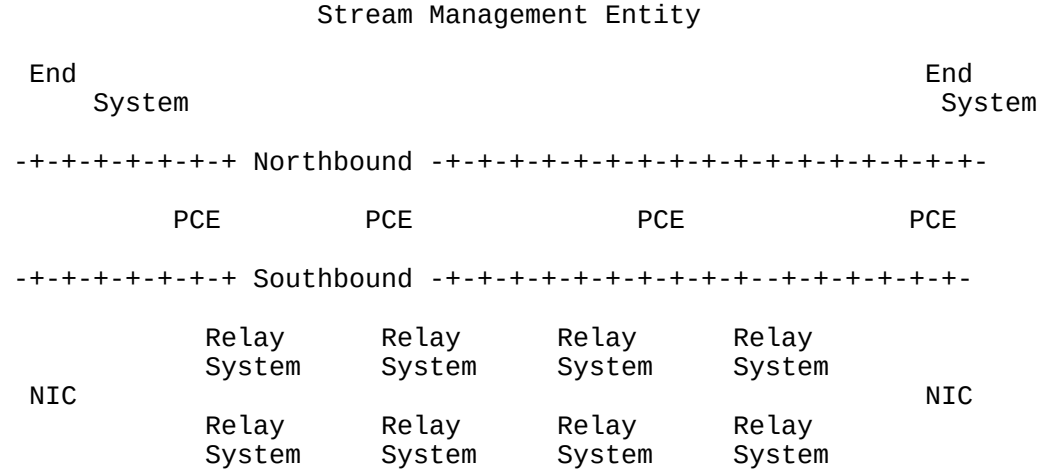


Figure 3

The relay systems (and eventually the end systems NIC) expose their capabilities and physical resources to the controller (the PCE), and update the PCE with their dynamic perception of the topology, across the Southbound Interface. In return, the PCE(s) set the per-stream paths up, providing a Stream Characterization that is more tightly coupled to the relay system Operation than a TSpec.

At the Network plane, relay systems exchange information regarding the state of the paths, between adjacent systems and eventually with the end systems, and forward packets within constraints associated to

each stream, or, when unable to do so, perform a last resort operation such as drop or declassify.

This specification focuses on the Southbound interface and the operation of the Network Plane.

4.4. Elements of DetNet Architecture

The DetNet architecture has a number of elements, discussed in the following sections:

a. A model for the definition, identification, and operation of DetNet streams (Section 4.5), for use by relay systems to classify and process individual packets following per-stream rules.

b. A model for the flow of data from an end system or through a relay system that can be used to predict the bounds for that system's impact on the QoS of a DetNet stream, without significantly constraining the method of implementing that system, for use by the Controllers to configure policing and shaping engines in Network Systems over the Southbound interface. The model includes:

    1. A model for queuing, transmission selection, shaping, preemption, and timing resources that can be used by an end system or relay system to control the selection of packets output on an interface. These models must have sufficiently well-defined characteristics, both individually and in the aggregate, to give predictable results for the QoS for DetNet packets (Section 4.7).

    2. A model for identifying misbehaving DetNet streams and mitigating their impact on properly functioning streams (Section 4.9).

c. A model for the relay system to inform the controller(s) of the information it needs for adequate path computations including:

    1. Systems' individual capabilities (e.g. can do replication, can do precise time).

    2. Link capabilities and resources (e.g. bandwidth, 0 delays, hardware deterministic support to the physical layer, ...)

    3. hysical resources (total and available buffers, timers, queues, etc)

4. Network Adjacencies (neighbors)

d. A model for the provision of a service, by end systems, or relay systems, to forward a DetNet stream over a simple or redundant path. The model includes:

   1. A model for an abstract relaying operation of either Routing or forwarding packets of a DetNet stream to a next-hop relay system, across Layer boundaries.

   2. A model of next-hop(s) information for replicating the packets of a DetNet stream, typically at or near the talker, merging and/or re-replicating those packets at other points in the network, and finally eliminating the duplicates, typically at or near the listener(s), in order to provide high availability (Section 3.3).

e. The protocol stack model for an end system and/or a relay system should support the above elements in a manner that maximizes the applicability of existing standards and protocols to the DetNet problem, allows for the creation of new protocols where needed, thus making DetNet an add-on feature to existing networks, rather than a new way to do networking. In particular this protocol stack supports networks in which the path from talker to listener(s) includes bridges and/or routers in any order (Section 4.10).

f. A variety of models for the provisioning of DetNet streams can be envisioned, including orchestration by a central controller or by a federation of controllers, provisioning by relay systems and end systems sharing peer-to-peer protocols, by off-line configuration, or by a combination of these methods. The provisioning models are similar to existing Layer-2 and Layer-3 models, in order to minimize the amount of innovation required in this area (Section 4.12).

4.5. DetNet streams

4.5.1. Talker guarantees

DetNet streams can by synchronous or asynchronous. The transmission of packets in synchronous DetNet streams uses time synchronization among the end and relay systems to control the flow of packets. Asynchronous DetNet streams are characterized by:

- A maximum packet size;

- An observation interval; and

- A maximum number of transmissions during that observation interval.

These parameters, together with knowledge of the protocol stack used (and thus the size of the various headers added to a packet), limit the number of bit times per observation interval that the DetNet stream can occupy the physical medium.

The talker promises that these limits will not be exceeded. If the talker transmits less data than this limit allows, the unused resources such as link bandwidth can be made available by the system to non-DetNet packets. However, making those resources available to DetNet packets in other streams would serve no purpose. Those other streams have their own dedicated resources, on the assumption that all DetNet streams can use all of their resources over a long period of time.

Note that there is no provision in DetNet for throttling streams; the assumption is that a DetNet stream, to be useful, must be delivered in its entirety. That is, while any useful application is written to expect a certain number of lost packets, the real-time applications of interest to DetNet demand that the loss of data due to the network is extraordinarily infrequent.

Although DetNet strives to minimize the changes required of an application to allow it to shift from a special-purpose digital network to an Internet Protocol network, one fundamental shift in the behavior of network applications that is impossible to avoid--the reservation of resources before the application starts. In the first place, a network cannot deliver finite latency and practically zero packet loss to an arbitrarily high offered load. Secondly, achieving practically zero packet loss for unthrottled (though bandwidth limited) streams means that bridges and routers have to dedicate buffer resources to specific streams or to classes of streams. The requirements of each reservation have to be translated into the parameters that control each system’s queuing, shaping, and scheduling functions and delivered to the hosts, bridges, and routers.

4.5.2. Incomplete Networks

The presence in the network of relay systems that are not fully capable of offering DetNet services complicates the ability of the relay systems and/or controller to allocate resources, as extra buffering, and thus extra latency, must be allocated at each point that is downstream from the non-DetNet relay system for some DetNet stream.

4.6. Data Flow Model through Systems

4.7. Queuing, Shaping, Scheduling, and Preemption

For this reason, the IEEE 802.1 Time-Sensitive Networking Task Group has defined a set of queuing, shaping, and scheduling algorithms that enable each bridge or router to compute the exact number of buffers to be allocated for each stream or class of streams.

4.8. Coexistence with normal traffic

A DetNet network supports the dedication of at least 75% of the network bandwidth to DetNet streams. But, no matter how much is dedicated for DetNet streams, It is z goal of DetNet to not interfere excessively with existing QoS schemes. It is also important that non-DetNet traffic not disrupt the DetNet stream, of course (see Section 4.9 and Section 6). For these reasons:

- Bandwidth (transmission opportunities) not utilized by a DetNet stream are available to non-DetNet packets (though not to other DetNet streams).

- DetNet streams can be shaped, in order to ensure that the highest-priority non-DetNet packet also is ensured a maximum latency.

- When transmission opportunities for DetNet streams are scheduled in detail, then the algorithm constructing the schedule should leave sufficient opportunities for non-DetNet packets to satisfy the needs of the uses of the network.

Ideally, the net effect of the presence of DetNet streams in a network on the non-DetNet packets is primarily a reductoin in the available bandwidth.

4.9. Fault Mitigation

One key to building robust real-time systems is to reduce the infinite variety of possible failures to a number that can be analyzed with reasonable confidence. DetNet aids in the process by providing filters and policers to detect DetNet packets received on the wrong interface, or at the wrong time, or in too great a volume, and to then take actions such as disabling the offending packet, shutting down the offending DetNet stream, or shutting down the offending interface.

It is also essential that filters and service remarking be employed to prevent non-DetNet packets from impinging on the resources allocated to DetNet packets.

There exist techniques, at present and/or in various stages of standardization, that can perform these fault mitigation tasks that deliver a high probability that misbehaving systemd will have zero impact on well-behaved DetNet streams, except of course, for the receiving interface(s) immediately downstream of the misbehaving device.

4.10. Protocol Stack Model

This section will be further developed. See [IEEE802.1CB], Annex C, for a description of the protocol stack. This is very much a work in progress, not a standard. See also [IEEE802.1Qcc].

4.11. Advertising resources, capabilities and adjacencies

4.12. Provisioning model

4.12.1. Centralized Path Computation and Installation

A centralized routing model, such as provided with a PCE (RFC 4655 [RFC4655]), enables global and per-stream optimizations. The model is attractive but a number of issues are left to be solved. In particular:

- whether and how the path computation can be installed by 1) an end device or 2) a Network Management entity,

- and how the path is set up, either by installing state at each hop with a direct interaction between the forwarding device and the PCE, or along a path by injecting a source-routed request at one end of the path.

4.12.2. Distributed Path Setup

Whether a distributed alternative without a PCE can be valuable should be studied as well. Such an alternative could for instance inherit from the Resource ReSerVation Protocol [RFC5127] (RSVP) flows.

In a Layer-2 only environment, or as part of a layered approach to a mixed environment, IEEE 802.1 also has work, either completed or in progress. [IEEE802.1Q-2011] Clause 35 describes SRP, a peer-to-peer protocol for Layer-2 roughly analogous to RSVP. Almost complete is [IEEE802.1Qca], which defines how ISIS can provide multiple disjoint paths or distribution trees. Also in progress is [IEEE802.1Qcc], which expands the capabilities of SRP.

5. Related IETF work

5.1. Deterministic PHB

[I-D.svshah-tsvwg-deterministic-forwarding] defines a Differentiated Services Per-Hop-Behavior (PHB) Group called Deterministic Forwarding (DF). The document describes the purpose and semantics of this PHB. It also describes creation and forwarding treatment of the service class. The document also describes how the code-point can be mapped into one of the aggregated Diffserv service classes [RFC5127].

5.2. 6TiSCH

Industrial process control already leverages deterministic wireless Low power and Lossy Networks (LLNs) to interconnect critical resource-constrained devices and form wireless mesh networks, with standards such as [ISA100.11a] and [WirelessHART].

These standards rely on variations of the [IEEE802154e] timeSlotted Channel Hopping (TSCH) [I-D.ietf-6tisch-tsch] Medium Access Control (MAC), and a form of centralized Path Computation Element (PCE), to deliver deterministic capabilities.

The TSCH MAC benefits include high reliability against interference, low power consumption on characterized streams, and Traffic Engineering capabilities. Typical applications are open and closed control loops, as well as supervisory control streams and management.

The 6TiSCH Working Group focuses only on the TSCH mode of the IEEE 802.15.4e standard. The WG currently defines a framework for managing the TSCH schedule. Future work will standardize deterministic operations over so-called tracks as described in [I-D.ietf-6tisch-architecture]. Tracks are an instance of a deterministic path, and the DetNet work is a prerequisite to specify track operations and serve process control applications.

[RFC5673] and [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 for improved reliability.

6. Security Considerations

Security in the context of Deterministic Networking has an added dimension; the time of delivery of a packet can be just as important as the contents of the packet, itself. A man-in-the-middle attack, for example, can impose, and then systematically adjust, additional delays into a link, and thus disrupt or subvert a real-time application without having to crack any encryption methods employed. See [RFC7384] for an exploration of this issue in a related context.

Furthermore, in a control system where millions of dollars of equipment, or even human lives, can be lost if the DetNet QoS is not delivered, one must consider not only simple equipment failures, where the box or wire instantly becomes perfectly silent, but bizarre errors such as can be coused by software failures. Because there is essentiall no limit to the kinds of failures that can occur, protecting against realistic equipment failures is indistinguishable, in most cases, from protecting against malicious behavior, whether accidental or intentional. See also Section 4.9.

Security must cover:

- the protection of the signaling protocol
- the authentication and authorization of the controlling systems
- the identification and shaping of the streams

7. IANA Considerations

This document does not require an action from IANA.

8. Acknowledgements

The authors wish to thank Jouni Korhonen, Erik Nordmark, George Swallow, Rudy Klecka, Anca Zamfir, David Black, Thomas Watteyne, Shitanshu Shah, Craig Gunther, Rodney Cummings, Wilfried Steiner, Marcel Kiessling, Karl Weber, Ethan Grossman and Pat Thaler, for their various contribution with this work.

9. Informative References

[AVnu] http://www.avnu.org/, "The AVnu Alliance tests and certifies devices for interoperability, providing a simple and reliable networking solution for AV network implementation based on the Audio Video Bridging (AVB) standards.", .

[CCAMP] IETF, "Common Control and Measurement Plane", <https://datatracker.ietf.org/doc/charter-ietf-ccamp/>.

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

[HSR-PRP] IEC, "High availability seamless redundancy (HSR) is a further development of the PRP approach, although HSR functions primarily as a protocol for creating media redundancy while PRP, as described in the previous section, creates network redundancy. PRP and HSR are both described in the IEC 62439 3 standard.", <http://webstore.iec.ch/webstore/webstore.nsf/artnum/046615!opendocument>.

[I-D.finn-detnet-problem-statement]
Finn, N. and P. Thubert, "Deterministic Networking Problem Statement", draft-finn-detnet-problem-statement-01 (work in progress), October 2014.

[I-D.ietf-6tisch-architecture]
Thubert, P., Watteyne, T., Struik, R., and M. Richardson, "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-architecture-05 (work in progress), January 2015.

[I-D.ietf-6tisch-tsch]
Watteyne, T., Palattella, M., and L. Grieco, "Using IEEE802.15.4e TSCH in an IoT context: Overview, Problem Statement and Goals", draft-ietf-6tisch-tsch-05 (work in progress), January 2015.

[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-03 (work in progress), March 2015.

[IEEE802.1AS-2011]
IEEE, "Timing and Synchronizations (IEEE 802.1AS-2011)", 2011, <http://standards.ieee.org/getIEEE802/download/802.1AS-2011.pdf>.

[IEEE802.1BA-2011]
IEEE, "AVB Systems (IEEE 802.1BA-2011)", 2011, <http://standards.ieee.org/getIEEE802/download/802.1BA-2011.pdf>.

[IEEE802.1CB]
IEEE, "Seamless Redundancy (IEEE Draft P802.1CB)", 2015, <http://p8021:go_wildcats@www.ieee802.org/1/files/private/cb-drafts/>.

[IEEE802.1Q-2011]
IEEE, "MAC Bridges and VLANs (IEEE 802.1Q-2011", 2011, <http://standards.ieee.org/getIEEE802/download/802.1Q-2011.pdf>.

[IEEE802.1Qat-2010]
IEEE, "Stream Reservation Protocol (IEEE 802.1Qat-2010)", 2010, <http://standards.ieee.org/getIEEE802/download/802.1Qat-2010.pdf>.

[IEEE802.1Qav]
IEEE, "Forwarding and Queuing (IEEE 802.1Qav-2009)", 2009, <http://standards.ieee.org/getIEEE802/download/802.1Qav-2009.pdf>.

[IEEE802.1Qca]
IEEE, "Path Control and Reservation", 2015, <http://p8021:go_wildcats@www.ieee802.org/1/files/private/ca-drafts/>.

[IEEE802.1Qcc]
IEEE, "Stream Reservation Protocol (SRP) Enhancements and Performance Improvements", 2015, <http://p8021:go_wildcats@www.ieee802.org/1/files/private/cc-drafts/>.

[IEEE802.1TSNTG]
IEEE Standards Association, "IEEE 802.1 Time-Sensitive Networks Task Group", 2013, <http://www.IEEE802.org/1/pages/avbridges.html>.

[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.

[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/IEC, "ISA100.11a, Wireless Systems for Automation, also IEC 62734", 2011, < http://www.isa100wci.org/en-US/Documents/PDF/3405-ISA100-WirelessSystems-Future-broch-WEB-ETSI.aspx>.

[ODVA] http://www.odva.org/, "The organization that supports network technologies built on the Common Industrial Protocol (CIP) including EtherNet/IP.", .

[PCE] IETF, "Path Computation Element", <https://datatracker.ietf.org/doc/charter-ietf-pce/>.

[Profinet]
http://us.profinet.com/technology/profinet/, "PROFINET is a standard for industrial networking in automation.", <http://us.profinet.com/technology/profinet/>.

[RFC2205] Braden, B., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.

[RFC5127] Chan, K., Babiarz, J., and F. Baker, "Aggregation of Diffserv Service Classes", RFC 5127, February 2008.

[RFC5673] Pister, K., Thubert, P., Dwars, S., and T. Phinney, "Industrial Routing Requirements in Low-Power and Lossy Networks", RFC 5673, October 2009.

[RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks", RFC 7384, October 2014.

[RFC7426] Haleplidis, E., Pentikousis, K., Denazis, S., Hadi Salim, J., Meyer, D., and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", RFC 7426, January 2015.

[TEAS] IETF, "Traffic Engineering Architecture and Signaling", <https://datatracker.ietf.org/doc/charter-ietf-teas/>.

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

Authors' Addresses

Norm Finn
Cisco Systems
170 W Tasman Dr.
San Jose, California 95134
USA

Phone: +1 408 526 4495
Email: nfinn@cisco.com

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

Phone: +33 4 97 23 26 34
Email: pthubert@cisco.com

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: September 5, 2015

C. Gunther, Ed.
HARMAN

March 4, 2015

Deterministic Networking Professional Audio Requirements
draft-gunther-detnet-proaudio-req-00

Abstract

This draft documents the needs in the Professional Audio industry to establish multi-hop paths and optional redundant paths for characterized flows with deterministic properties.

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 September 5, 2015.

Table of Contents

1. Introduction

Professional Audio (Pro-A) includes the simple and small network used by a garage band which may contain a handful of devices, as well as the large theme park spread across 25,000 acres or more. It is worth noting that these theme parks may exist on multiple continents and share content around the world.

Some examples of Pro-A networks include:

- Garage bands
- Portable PA
- Churches

- Concert halls
- Recording and broadcasting studios
- Cinema and theater sound
- Train stations
- Stadiums
- Airports

While many of these uses have common requirements there are some unique usage models that will be highlighted in this document.

2. Requirements Language

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

3. Stream Characteristics

All streams of interest to the Pro-A world have the same requirements related to establishing a path and allocating bandwidth as any other type of network application. This section of the draft is meant to introduce other concerns associated with streams in a Pro-A network.

3.1. Emergency Notifications

Audio systems installed in public environments have unique requirements with regards to health, safety and fire concerns. For example [ISO7240-16] subjects equipment to tests that can simulate an emergency situation. The purpose of this section is to provide a very basic set of requirements that an underlying network must provide if it is to be used in public areas. It would be advantageous to establish a liaison with the International Standards Organization (ISO) so that the referenced ISO 7240 standards could be made available for Deterministic Networking (DetNet) review for the specific details.

The remainder of this section is simply a synopsis of some of the requirements found in the ISO 7240 standard. The wording in that standard supersedes anything specified in this section and it should be referenced for the specific requirements.

Any numbers in this section surrounded by braces refers to the specific section within ISO 7240-16:2007 (for example {7.1.1} is a reference to section 7.1.1).

One such requirement is a maximum of 3 seconds {7.1.1} for a system to respond to an emergency detection and begin sending appropriate warning signals and alarms. When these conditions occur the audio system must be able to disable normal functions {7.1.4} not associated with emergency functionality, without the need for human intervention.

Announcements must be able to be made within 20 seconds of a system reset {7.9.2.2}.

In the event of equipment failure the backup equipment must be able to take over within 10 seconds {14.4.1}. This would include detection time, new path configuration, etc.

3.2. Content Protection

Digital Rights Management (DRM) is very important to the Pro-A and Professional Video industries. Any time that protected content is introduced into a network there are DRM concerns that must be maintained. (See [CONTENT_PROTECTION]).

As an example, two techniques are Digital Transmission Content Protection (DTCP) and High-Bandwidth Digital Content Protection (HDCP). HDCP content is not approved for retransmission within any other type of DRM, while DTCP may be retransmitted under HDCP. Therefore if the source of a stream is outside of the network and it uses HDCP protection it is only allowed to be placed on the network with that same HDCP protection.

3.3. Multiple Sinks

Pro Audio systems often have multiple sinks (e.g.: speakers) connected to a single source. In order to keep bandwidth utilization of shared links to a minimum multicast addressing is commonly used.

3.4. Super Stream = Two or More Serial Streams

Audio content delivered from a source (e.g.: microphone or guitar) can be sent through one or more stages of processing before it reaches the sink(s). For example, one stream may be used to send audio from a microphone hub to a digital processor that will match the singers pitch to that of a guitar. A second stream will then take that processed audio to a mixing console. A third stream is then required to move the mixed audio to an amplified speaker. Not

only does this one super "stream" require three physical streams to be created, but the overall latency of all three streams plus the digital processing at each hop must not exceed 10-15 msec. See slide 6 of [SRP_LATENCY].

3.5. Unused Reservations and Best-Effort Traffic

Often times reservations are created, but not used until some time later in a live show. This is really more of a comfort issue for the show's producers; they just want to know that there is no reason an important reservation's request could be refused during a live performance.

In other situations a single reservation may be used for different content at different times throughout the day. It is convenient to create a single reservation that is large enough for the biggest bandwidth consumer although that could be wasteful on smaller streams.

In both these cases it is advantageous for other best-effort traffic to be able to use that unused bandwidth so that the full bandwidth of the network can be utilized at all times. This best-effort traffic could consist of "meter data" which helps an operator understand what is going on at the other end of Pro-A system in an amusement park. Or it could be used for file transfers or venue updates. Regardless of the reason, Pro-A installations will want to be able to use any reserved bandwidth that is unused.

3.6. Maximum and Acceptable Latency

In order to synchronize speakers throughout a venue it is critical for each sink (amplified speaker) to know what the maximum latency is it can expect to see from the network. That maximum latency from each sink is sent back to the source, or an associated Controller, so the presentation time of the Pro-A audio data samples can be set. In addition, sinks that are fewer hops away from the source will know how much memory they will need to provide in order to buffer the content that will be presented at some later time.

A Controller may also collect the various maximum latency numbers and decide to exclude the sinks that are too many hops away since they will place unrealistic buffering requirements on the sinks that are very few hops from the source.

Additionally, sinks that are closer to the source can inform the network that they can accept more latency than the network is currently offering since they will be buffering packets to match play-out time of father away sinks. This acceptable latency can be

used by the network to move a reservation on a short path to a longer path in order to free up bandwidth for other critical streams on that short path. See slides 3-5 of [SRP_LATENCY].

3.7. Latency Per Sink

As previously mentioned a single stream may be sent to multiple sinks. This use case introduces the concept of more stringent latency requirements for some sinks, whereas other sinks have more flexible latency requirements. A live outdoor concert has stringent requirements for delivering the audio to the speaker systems, yet can have very flexible requirements for that same audio content that is delivered to a mobile recording studio that is set up nearby. See slide 7 of [SRP_LATENCY].

3.8. Layer 3 Interconnecting Layer 2 Islands

The DetNet solution for Layer 3 networks should support Layer 3 segments that can connect to Layer 2 networks that do not support Layer 3 protocols.

3.9. Link Aggregation

If any type of link aggregation is proposed as part of the DetNet solution there must be a technique used that can determine the maximum latency that a packet may experience when flowing across any links in that aggregation.

Or, an alternative could be to report the maximum latency of a single link within the link aggregation and then enforce that the stream will only use that link when establishing the path.

3.10. Layer 3 Multicast

Because of the MAC Address forwarding nature of Layer 2 bridges it is important that a multicast MAC Address is only associated with one stream. This will prevent reservations from forwarding packets from one stream down a path that has no interested sinks simply because there is another stream on that same path that shares the same multicast MAC address.

Since each multicast MAC Address can represent 32 different IPv4 multicast addresses there must be a process put in place to make sure this does not occur. Optionally it could be stated that Deterministic Networking will recommend the use of IPv6, although the impact of such a decision upon existing IPv4 installations should be discussed.

3.11. Segregate Traffic

Sink devices may have limited processing power. In order to not overwhelm the CPUs in these devices it is important to limit the amount of traffic that these devices must process. Packet forwarding rules should eliminate extraneous streaming traffic from reaching these devices; however there may be other types of broadcast traffic that should be eliminated where possible. This is often done by VLANs or IP subnets.

3.12. Elapsed Time to Build a Reservation

During a venue change in a show various modifications to reservations may be required. Some existing reservation may be torn down and other reservations may be established. On the Pro-A side this may be a simple reconfiguration of the speakers so the sound field can be created in a different way, or inclusion or exclusion of certain areas in the physical environment.

When video is added to the mix this may be switching from one camera to another. Currently video systems use expensive switching hardware to switch inputs at the head-end of the final feed. Interest has been expressed from the Broadcast industry to the IEEE AVB group for using the network as the video switch (see [STUDIO_IP]).

There is also the issue of the time between power-on and establishment of the first set of reservations. In many situations the appropriate thing to do is simply reestablish all paths and bandwidth reservations as were in place when the power was turned off, doing this as quickly as possible. This is particularly true when recovering from a power failure, or accidental removal of an Ethernet cable or power cord.

4. Use Cases

4.1. Singularity of IT and AV Networks

A recent large installation of a Pro-A network based on IEEE 802.1 AVB technology encompassed a 194,000 sq ft, $125 million facility. The network is capable of handling 46 Tbps of throughput with 60,000 simultaneous signals. Inside the facility are 1,100 miles of fiber feeding four audio control rooms. Phase I of this project was for audio, the next phase will include video as well. One of the future goals of this project is to have the capability to integrate IT infrastructure with the audio streaming technology. Details of this installation can be found here [ESPN_DC2].

4.2. Combining Local and Remote Content

One advantage of a guaranteed reservation with a small bounded latency is the reduced buffering requirements on sink devices. As mentioned earlier there are large theme parks, megachurches, and other venues that wish to broadcast a live event from one physical location to another physical location. These may be across town or across the globe and the content would be delivered via a layer 3 protocol. Depending on the technology available, latency bounds and jitter caused by Internet delivery of content can have a huge impact on the buffering requirements at the receiving site.

In these situations it is acceptable at the local location for content from the live remote site to be delayed (buffered) a reasonable amount to allow for a statistically acceptable amount of latency in order to reduce jitter. However, once the content begins playing in the local location any audio artifacts caused by the local network are unacceptable, especially in those situation where a live local performer is "mixed" into the feed from the remote location.

With these scenarios a single gateway device at the local network that is receiving the feed from the remote site would provide the expensive buffering required to mask the latency and jitter issues associated with long distance delivery. Sink devices in the local location would have no additional buffering requirements, and thus no additional costs, beyond those required for delivery of local content. The sink device would be receiving the identical packets as those sent by the source and would be unaware that there were any latency or jitter issues along the path.

4.3. Lots of Small Devices

Consumers expect more and more from their theater experiences. One example is the use of individual theater seat speakers and effects systems. In order to be cost effective these systems must be inexpensive per seat since the quantities in a single theater can reach hundreds or thousands of seats.

Discovery protocols alone in a one thousand seat theater can generate a lot of broadcast traffic that can put an unnecessary load on a low powered CPU. An installation like this will require some type of traffic segregation that can create groups of seats to reduce traffic within that group. All seats in the theater must still be able to communicate with a central controller.

5. Acknowledgements

The editor would like to acknowledge the help of the following individuals and the companies they represent:

Jeff Koftinoff, Meyer Sound

Jouni Korhonen, Associate Technical Director, Broadcom

Pascal Thubert, CTAO, Cisco

Kieran Tyrrell, Sienda New Media Technologies GmbH

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

7.1. Content Protection

As mentioned earlier any solutions that would be recommended for the Professional A/V space must support DRM.

7.2. Denial of Service

Many industries that are moving from the analog wire world to the digital network world have little understanding of the pitfalls that they can create for themselves by an improperly installed system. DetNet should consider ways to provide security against DoS attacks in solutions directed at these markets.

One example this author is aware of involved the use of technology that allows a presenter to "throw" the content from their tablet or smart phone onto the A/V system that is then viewed by all those in attendance. The facility introducing this technology was quite excited to allow such modern flexibility to those who came to speak. One thing they hadn’t realized was that since no security was put in place around this technology it left a hole in the system that allowed other attendees to "throw" their own content onto the A/V system.

7.3. Control Protocols

Pro-A systems can include amplifiers that are capable of generating several hundreds or thousands of watts of audio power. If used incorrectly these systems can cause hearing damage to those in the vicinity of the speaker arrays. The traffic that controls these

devices must be protected and that is mostly a concern of those providing that service. However, the configuration protocols that create the network paths used by the Pro-A traffic should be protected as well so that high-volume content cannot be sent to areas that are not meant to receive it.

8. References

8.1. Normative References

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

8.2. Informative References

[CONTENT_PROTECTION]
Olsen, D., "1722a Content Protection", 2012, <http://grouper.ieee.org/groups/1722/contributions/2012/avtp_dolsen_1722a_content_protection.pdf>.

[ESPN_DC2]
Daley, D., "ESPN’s DC2 Scales AVB Large", 2014, <http://sportsvideo.org/main/blog/2014/06/espns-dc2-scales-avb-large>.

[ISO7240-16]
ISO, "ISO 7240-16:2007 Fire detection and alarm systems -- Part 16: Sound system control and indicating equipment", 2007, <http://www.iso.org/iso/catalogue_detail.htm?csnumber=42978>.

[SRP_LATENCY]
Gunther, C., "Specifying SRP Latency", 2014, <http://www.ieee802.org/1/files/public/docs2014/cc-cgunther-acceptable-latency-0314-v01.pdf>.

[STUDIO_IP]
Mace, G., "IP Networked Studio Infrastructure for Synchronized & Real-Time Multimedia Transmissions", 2007, <http://www.ieee802.org/1/files/public/docs2047/avb-mace-ip-networked-studio-infrastructure-0107.pdf>.

Author’s Address

Craig Gunther (editor)
Harman International
10653 South River Front Parkway
South Jordan, UT 84095
USA

Phone: +1 801 568-7675
Email: craig.gunther@harman.com
URI: http://www.harman.com

6TiSCH
Internet-Draft
Intended status: Informational
Expires: September 10, 2015

Q. Wang, Ed.
Univ. of Sci. and Tech. Beijing
X. Vilajosana
Universitat Oberta de Catalunya
T. Watteyne
Linear Technology
March 9, 2015

6TiSCH Operation Sublayer (6top) Interface
draft-ietf-6tisch-6top-interface-03

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 of commands for the internal use of the 6top sublayer and/or to be used by implementers as an API guideline or basic specification.

Table of Contents

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. Requirements Language

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

3. 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

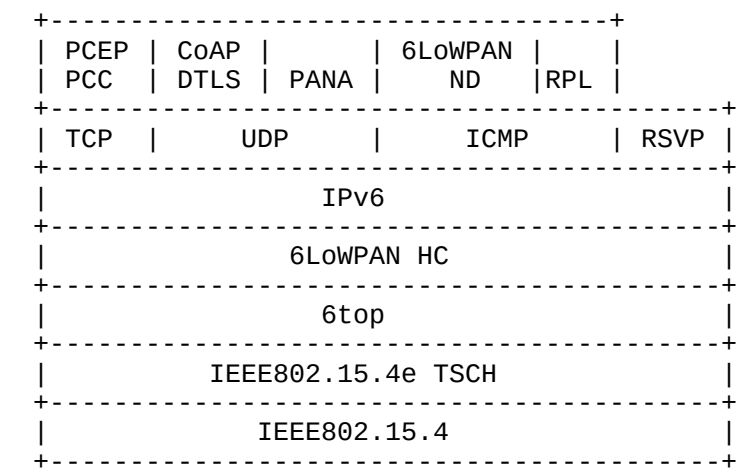


Figure 1

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 3.1.

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 3.2.

Section 4 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 5. This set of commands is designed to support decentralized, centralized and hybrid scheduling solutions.

3.1. 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.

3.1.1. 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.

3.1.2. 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].

3.2. 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

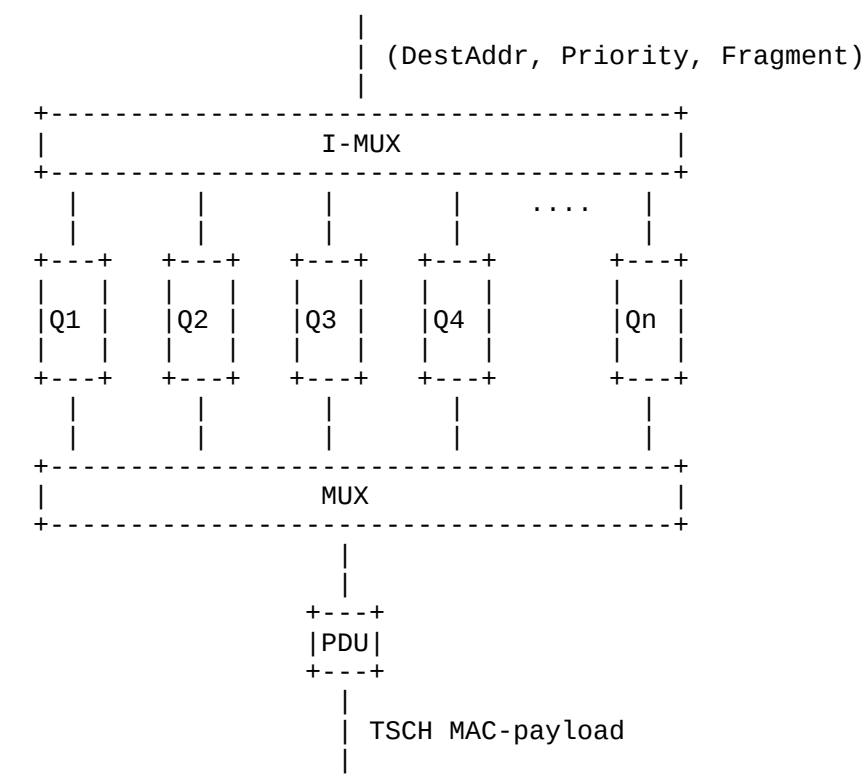


Figure 2

In Figure 2, Qi 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 5), 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.

4. Generic Data Model

This section presents the generic data model of the 6top sublayer, using the YANG data modeling langage. 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].

4.1. YANG model of the 6top MIB

```
typedef nodeaddresstype {
    type uint64;
    description
       "The type to store node address. It can be 64bits EUI address;
       or the short address defined by 6top, constrained by TSCH
       macNodeAddress size, 2-octets. If using TSCH as MAC, the
       higher 6-octets should be filled with 0, and lowest 2-octets
       is neighbor short address";
 }


 list CellList {
       key "CellID";
       min-elements 1;
       unique "SlotframeID SlotOffset ChannelOffset";
       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 leafref {
               path "/SlotframeList/SlotframeID";
            }
            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 uint16;
            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 NodeAddress {
             type nodeaddresstype;
             description
             "specify the target node address.";
          }
          leaf TrackID {
             type leafref {
                path "/TrackList/TrackId";
             }
             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 {
                mandatory true;
                type uint8;
                description
                "Number of statistics collected on the cell";
             }
             list MeasureList {
                key "StatisticsMetricsID";
                min-elements 1;

                leaf StatisticsMetricsID{
                   type leafref {
                      path "/StatisticsMetricsList/StatisticsMetricsID";
                   }
                   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";
         min-elements 1;
         description
         "List of all of the slotframes used by the node.";

         leaf SlotframeID {
            mandatory true;
            type uint8;
            description
            "Equal to SlotframeHandle defined in TSCH";
            reference
            "IEEE802154e";
         }
         leaf NumOfSlots {
            mandatory true;
            type uint16 {
               range "1..max";
            }
            description
            "indicates how many timeslots in the slotframe";
         }
      }

    list Version {
       key "Version";
       description
       "Provides a unique identification for the set of resources
           defined in this draft. Provides a major and minor version
           number that may be accessible independently";

           leaf major {
          type uint8;
       }
       leaf minor {
          type uint8;
       }
     }

       list MonitoringStatusList {
          key "MonitoringStatusID";
          min-elements 1;
          unique "SlotframeID NodeAddress";
          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 leafref {
                path "/SlotframeList/SlotframeID";
             }
             description
             "SlotframeID, one in SlotframeList, indicates the slotframe
             being monitored";
             reference
             "IEEE802154e";
          }
          leaf NodeAddress {
             type nodeaddresstype;
          }
          leaf EnforcePolicy {
             type enumeration {
                enum DISABLE;
                enum BESTEFFORT;
                enum STRICT;
                enum OVERPROVISION;
             }
             default DISABLE;
             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;
             must "../EnforcePolicy < > DISABLE ./";
             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";
  min-elements 1;
  unique "SlotframeID SlotOffset ChannelOffset NodeAddress";
  description
  "List of Statistics Metrics used in the node.";

  leaf StatisticsMetricsID {
         type uint16;
  }
  leaf SlotframeID {
         type leafref {
                path "/SlotframeList/SlotframeID";
         }
         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 uint16;
         description
         "Specific channelOffset to which the statistics metrics applies
         to. If empty, applies to all channels";
         reference
         "IEEE802154e";
  }

  leaf NodeAddress {
         type nodeaddresstype;
         description
         "If NodeAddress is empty, applies to all neighbor nodes.";
  }

  leaf Metrics {
  type enumeration {
         enum macCounterOctets
         enum macRetryCount
         enum macMultipleRetryCount
         enum macTXFailCount
         enum macTXSuccessCount
         enum macFCSErrorCount
         enum macSecurityFailure
         enum macDuplicateFrameCount
         enum macRXSuccessCount
         enum macNACKcount
         enum PDR;
         enum ETX;
         enum RSSI;
         enum LQI;
  }
  description
  "The metric to be monitored. Include those provided by underlying
  IEEE 802.15.4e TSCH -- see table 4i (2012).
  PDR,ETX,RSSI,LQI are maintained by 6top. ";
 }
  leaf Window {
         type uint16;
         description
         "measurement period, in Number of the slotframe size";
  }
  leaf Enable {
         type enumeration {
                enum DISABLE;
                enum ENABLE;
         }
```

```
          default DISABLE;
          description
          "indicates the StatisticsMetric is active or not";
    }
}
```

```
list EBList {
         key "EbID";
         min-elements 1;
         description
         "List of information related with the EBs used by the node";

         leaf EbID {
                type uint8;
         }
         leaf CellID {
                type leafref {
                   path "/CellList/CellID";
                }
                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;
          }
          default 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 NodeAddress {
          type nodeaddresstype;
          description
          "Specifies the address of selected time source neighbors.";
   }

   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 "NodeAddress";
         description
         "statistics per communication link. ";

         leaf NodeAddress {
            type nodeaddresstype;
            description
            "Specifies the address of the 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";
  min-elements 1;
  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 leafref {
                  path "/TrackList/TrackId";
               }
               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 leafref {
                   path "/CellList/CellID";
                }
                description
                "The CellID, indicates the Rx cell on which the packet will
                come in.";
         }
  }
  list OutputCellIds {
         key "CellID";
         leaf CellID{
                type leafref {
                   path "/CellList/CellID";
                }
                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";
   min-elements 1;
   unique "TrackOwnerAddr InstanceID";
   description
   "List of the tracks through the node. At lease the best effort
   track is existing";

   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 leafref {
                     path "/SlotframeList/SlotframeID";
              }
              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 uint16;
              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 leafref{
                     path "/ChunkList/ChunkId";
              }
              description
              "Identifier of the chunk the cell belongs to";
           }
           leaf CellID{
              type leafref {
                     path "/CellList/CellID";
              }
              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;
              }
           }
    }
```

4.2. 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 comannds.

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";
           min-elements 1;
           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";
      min-elements 1;
      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.";
       }
   }
```

4.3. Yang Model for the Security aspects of 6top

The [I-D.ietf-6tisch-architecture] and [I-D.richardson-6tisch--security-6top] define the attributes needed to secure network bootstraping and joining and authentication processes. The following attributes are exposed by 6top interface to enable access and configuration to the security mechanisms carried out by 6top management entity.

```
 container SecurityAttributes{

    leaf SecurityMode {
       type enumeration {
          enum NO_SECURITY;
          enum NETWORK_WIDE_MIC;
          enum NETWORK_WIDE_DHE_PSK;
          enum NETWORK_WIDE_IKE2_PSK;
          enum PK_DTLS_ECDSA;
          enum PK_IKEv2_ECDSA;
```

```
          enum OTHER;
       }
    description
     "The security mode is to be used.";
    }

    leaf-list Certificate{
       type uint8;
       min-elements 128;
       description "A list of bytes for the
                 certificate ECDSA PKIX or PSK";
    }

    leaf DevID {
       type enumeration {
            enum IDevID;
            enum LDevID;
          }
       description " indicate the feature of DevID.";
    }

    leaf-list PSK{
       type uint8;
       min-elements 128;
       description "A list of bytes for the PSK while using PSK method";
    }

    leaf PanID {
       type uint16;
       description "2 Bytes the network PANID";
    }

    leaf JoinAssistant {
       type enumeration {
          enum TRUE;
          enum FALSE;
       }
       description "a toggle which enables a node to
                 become a join assistant.";
    }

    leaf-list ULA{
       type uint8;
       min-elements 16;
       description "A ULA to be announced in the
                    RA for joining nodes. It is 128bits+prefixlen.
                    A device with multiple interfaces
                    should configure different 64-bit prefixes.";
```

```
    }

    leaf BeaconWellKnownKey{
       type string;
       default "6TISCHJOIN";
       description "the well known beacon key";
    }

    leaf-list JCEAddress{
       type uint8;
       min-elements 8;
       description "the address of the JCE,
                 for the ACL about
                 who can contact joining nodes.";
    }
 }
```

5. 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 4.1).

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 4.1).

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 4.1).

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 4.1) will be modified.

DELETE.hardcell: To remove a hard cell. This removes the hard cell from the node's schedule, from CellList (Section 4.1).

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 4.1) 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 modificaition of CellLists (Section 4.1) on both sides.

CREATE.slotframe: Creates a new slotframe. Adds a entry to the SlotframeList (Section 4.1).

READ.slotframe: Returns the information of a slotframe given its slotframeID from SlotframeList (Section 4.1).

UPDATE.slotframe: Change the number of timeslots in a slotframe given its slotframeID in SlotframeList (Section 4.1).

DELETE.slotframe: Deletes a slotframe, remove it from SlotframeList (Section 4.1).

CONFIGURE.monitoring: Configures the level of QoS the Monitoring process MUST enforce, i.e. config MonitoringStatusList (Section 4.1).

READ.monitoring: Reads the current Monitoring status from MonitoringStatusList (Section 4.1).

CONFIGURE.statistics: Configures the statistics process in StatisticsMetricsList(Section 4.1). 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 4.1).

RESET.statistics: Resets the gathered statistics in CellList (Section 4.1).

CONFIGURE.eb: Configures EBs, i.e. configures EBlist (Section 4.1).

READ.eb: Reads the EBs configuration from EBList (Section 4.1).

CONFIGURE.timesource: Configures the Time Source Neighbor selection process, i.e. configure TimeSource (Section 4.1).

READ.timesource: Retrieves information about the time source neighbors of that node from TimeSource (Section 4.1).

CREATE.neighbor: Creates an entry for a neighbor in the neighbor table, i.e. NeighborList (Section 4.1).

READ.all.neighbor: Returns the list of neighbors of that node according to NeighborList (Section 4.1).

READ.neighbor: Returns the information of a specific neighbor of that node specified by its neighbor address according to NeighborList (Section 4.1).

UPDATE.neighbor: Updates the last status for a given TargetNodeAddress in the NeighborList (Section 4.1).

DELETE.neighbor: Deletes a neighbor given its address from NeighborList (Section 4.1).

CREATE.queue: Creates and Configures a queue in QueueList (Section 4.1).

READ.queue: Reads the queue configuration for given QueueId from QueueList (Section 4.1).

READ.queue.stats: For a given QueueId, reads the queue statistics information from the QueueList (Section 4.1).

UPDATE.queue: For a given QueueId, update its configuration in the QueueList (Section 4.1).

DELETE.queue: Deletes a Queue for a given QueueId from the QueueList (Section 4.1).

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 4.1).

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 4.1).

CREATE.chunk: Creates a chunk which consists of one or more unscheduled cells, i.e. add an entry to the ChunkList (Section 4.1).

READ.chunk: Returns the information of a chunk given its ChunkID from ChunkList (Section 4.1).

DELETE.chunk: For given ChunkId, removes a chunk from the ChunkList (Section 4.1), which also causes all of the scheduled cells in the chunk to be deleted from the TSCH schedule and CellList (Section 4.1).

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 4.1). In addition, 6top will change the attributes corresponding to the cell in the ChunkCellList (Section 4.1), 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 4.1).

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 4.1). In addition, it changes the attributes corresponding to the cell in the ChunkCellList (Section 4.1), i.e. its CellID is changed back to 0xFFFF, and its Status is changed to UNSCHEDULED.

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.

[I-D.ietf-6tisch-tsch]
Watteyne, T., Palattella, M., and L. Grieco, "Using IEEE802.15.4e TSCH in an IoT context: Overview, Problem Statement and Goals", draft-ietf-6tisch-tsch-05 (work in progress), January 2015.

[I-D.ietf-6tisch-architecture]
Thubert, P., Watteyne, T., Struik, R., and M. Richardson, "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-architecture-06 (work in progress), March 2015.

[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-03 (work in progress), January 2015.

[I-D.ietf-6tisch-minimal]
Vilajosana, X. and K. Pister, "Minimal 6TiSCH Configuration", draft-ietf-6tisch-minimal-06 (work in progress), March 2015.

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

[I-D.ietf-6tisch-coap]
Sudhaakar, R. and P. Zand, "6TiSCH Resource Management and Interaction using CoAP", draft-ietf-6tisch-coap-02 (work in progress), December 2014.

[I-D.richardson-6tisch--security-6top]
Richardson, M., "6tisch secure join using 6top", draft-richardson-6tisch--security-6top-04 (work in progress), November 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

6TiSCH
Internet-Draft
Intended status: Standards Track
Expires: September 10, 2015

P. Thubert, Ed.
Cisco
T. Watteyne
Linear Technology
R. Struik
Struik Security Consultancy
M. Richardson
SSW
March 9, 2015

An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4e
draft-ietf-6tisch-architecture-06

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.

Table of Contents

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 per device compared to traditional wired technology, at any distance ranging from Near Field to interplanetary, and in circumstances where wiring may not appear practical, for instance on 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, quite sensitive to latency, and with a high degree of operational criticality so that loss should be minimized at all times. Such traffic is not limited to professional Audio/ Video networks, but is also found in command and control operations such as industrial automation and vehicular sensors and actuators.

At IEEE802.1, the Audio/Video Task Group [IEEE802.1TSNTG] Time Sensitive Networking (TSN) to address Deterministic Ethernet. The IEEE802.15.4 Medium access Control (MAC) has evolved with the new IEEE802.15.4e TimeSlotted Channel Hopping (TSCH) [I-D.ietf-6tisch-tsch] mode for deterministic industrial-type applications.

Though at a different time scale, both TSN and TSCH 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 Wi-Fi emitters).

This document is the first volume of 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) [PCE], in a distributed fashion using the Routing Protocol for Low Power and Lossy Networks (RPL) [RFC6550], or in a mixed mode. The Backbone Routers may perform proxy IPv6 Neighbor Discovery (ND) [RFC4861] operations over the backbone on behalf of the wireless devices (also called motes), so they can share a same IPv6 subnet and appear to be connected to the same backbone as classical devices. The Backbone Routers may alternatively redistribute the registration in a routing protocol such as OSPF [RFC5340] or BGP [RFC2545], or inject them in a mobility protocol such as MIPv6 [RFC6275], NEMO [RFC3963], or LISP [RFC6830].

TimeSlots and other device resources are managed by an abstract Network Management Entity (NME), which may cooperate with the PCE in order to minimize the interaction with and the load on the constrained device.

Hints are provided on a security framework that will be completed in the round of this document.

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] where the 6LoWPAN Router (6LR) and the 6LoWPAN Border Router (6LBR) are introduced, 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], [RFC4080], and [RFC5191].

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

Some aspects of this architecture derive from existing industrial standards for Process Control such as ISA100.11a [ISA100.11a]and WirelessHART [WirelessHART], by its focus on Deterministic Networking, in particular with the use of the IEEE802.15.4e [IEEE802154e] TSCH MAC and a 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. In such applications, Deterministic Networking applies mainly to control loops and movement detection, but it can also be used for supervisory control flows and management.

An incremental set of industrial requirements is 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, as discussed in [I-D.ietf-roll-rpl-industrial-applicability].

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

The scope of the present work is a subnet that, in its basic configuration, is made of a TSCH [I-D.ietf-6tisch-tsch] MAC Low Power Lossy Network (LLN).

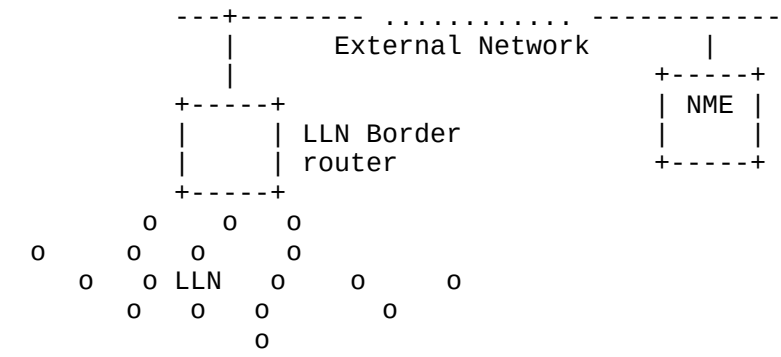


Figure 1: Basic Configuration of a 6TiSCH Network

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 [RFC6775] (6LoWPAN ND). RPL [RFC6550] enables routing within the LLN, in the so called Route Over fashion, either in storing (stateful) or non-storing (stateless, with routing headers) mode.

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 Border Router for the LLN to the outside. The LBR is usually powered. More on RPL Instances can be found in section 3.1 of RPL [RFC6550], in particular "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]. This architecture suggests new work to standardize the participation of non-RPL leaves and the registration to backbone routers for proxy operations. For instance, the registration backbone could be based on 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 process on joining or re-joining the network is discussed in Section 13 and the applicability of existing protocols such as the Protocol for Carrying Authentication for Network access (PANA) [RFC5191] will be studied in a next volume of this document.

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.

In the extended configuration, a Backbone Router (6BBR) acts as an Energy Aware Default Router (NEAR) as defined in [I-D.chakrabarti-nordmark-6man-efficient-nd]. The 6BBR performs ND proxy operations between the registered devices and the classical ND devices that are located over the backbone. 6TiSCH 6BBRs synchronize with one another over the backbone, so as to ensure that the multiple LLNs that form the IPv6 subnet stay tightly synchronized.

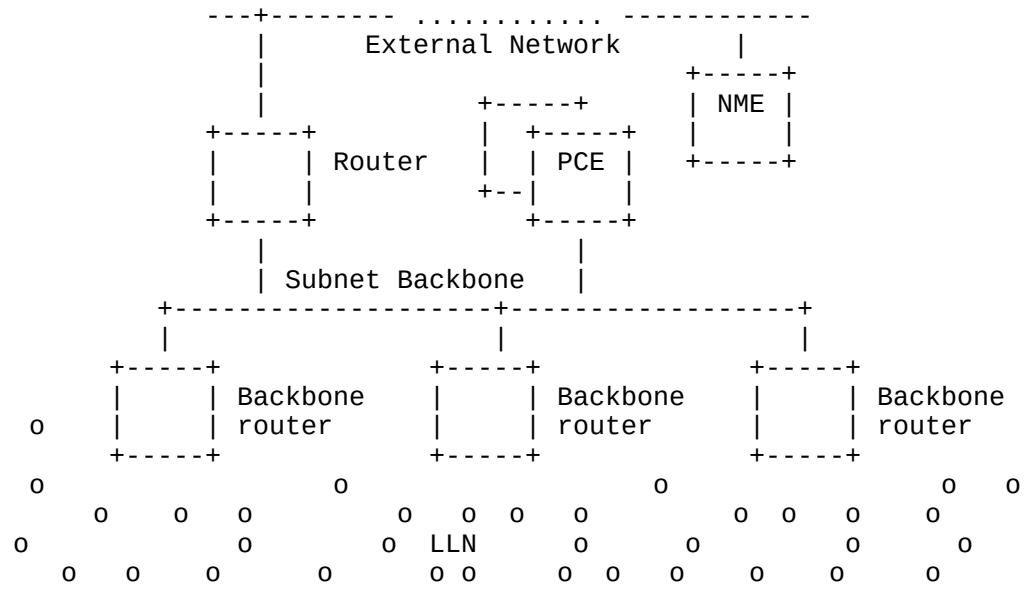


Figure 2: Extended Configuration of a 6TiSCH Network

In order to serve nodes that are multiple hops away, an integrated RPL root and 6LBR may be collocated with the 6BBR, or attached to the 6BBR in which case they would perform the registration on behalf of the remote LLN nodes - they proxy the efficient ND registration over

the LLN in order for the 6BBR to perform proxy ND operations over the backbone.

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. Note: A DetNet - for Deterministic Networking - Mailing List was formed at the IETF to study Layer-3 aspects of the technology, and cover networks that span multiple Layer-2 domains.

5. Scope

5.1. Components

In order to control the complexity and the size of the 6TiSCH work, the architecture and the associated IETF work are staged in volumes. This document covers the first stage of the work, as specified by the WG charter. If the work continues as expected, further volumes will complete this piece and provide the full coverage of IPv6 over TSCH.

The main architectural blocks are represented below to help detail what is covered and what is not yet covered from the global 6TiSCH architecture by this initial volume:

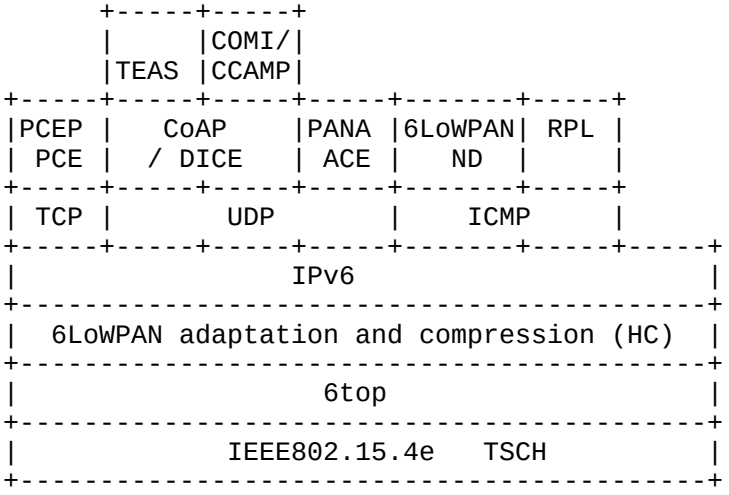


Figure 3: Envisioned 6TiSCH protocol stack

RPL is the routing protocol of choice for LLNs. So far, there was no identified need to define a 6TiSCH specific Objective Function. The Minimal 6TiSCH Configuration [I-D.ietf-6tisch-minimal]describes the operation of RPL over a static schedule used in a slotted aloha

fashion, whereby all active slots may be used for emission or reception of both unicast and multicast frames. The architecture of the operation of RPL over a dynamic schedule is deferred to a subsequent volume of the architecture.

6TiSCH has adopted the general direction of CoAP Management Interface (COMI) [I-D.vanderstok-core-comi] for the management of devices. This is leveraged for instance for the implementation of the generic data model for the 6top sublayer management interface [I-D.ietf-6tisch-6top-interface]. The proposed implementation is based on CoAP and CBOR, and specified in 6TiSCH Resource Management and Interaction using CoAP [I-D.ietf-6tisch-coap]. At the time of this writing, COMI and the dependent specifications are still work in progress at this time, and DTLS In Constrained Environments (DICE) [DICE] is the probable way LLN nodes will provide end-to-end security for UDP/CoAP packets.

The work on centralized track computation is deferred to a subsequent volume of the architecture. The Path Computation Element (PCE) is certainly the core component of that architecture. Around the PCE, a protocol such as an extension to a TEAS [TEAS] protocol (maybe running over CoAP as illustrated) will be required to expose the device capabilities and the network peers to the PCE, and a protocol such as a lightweight PCEP or an adaptation of CCAMP [CCAMP] G-MPLS formats and procedures will be used to publish the tracks, computed by the PCE, to the devices (maybe in a fashion similar to RSVP-TE).

There is a debate whether PANA (Layer-3), IEEE802.1x (Layer-2) or some light weight variation of those 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. This piece of the architecture is also deferred to a subsequent volume of the architecture. A status of the work can be found in Section 13.

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. The work on the operations of that layer, in particular related to dynamic scheduling, is only introduced here, and should be detailed further in a subsequent volume of the architecture.

5.2. Dependencies

At the time of this writing, the components and protocols that are required to implement this stage of architecture are not fully available from the IETF. In particular, the requirements on an evolution of 6LoWPAN Neighbor Discovery that are needed to implement the Backbone Router as covered by this stage of the architecture are detailed in [I-D.thubert-6lo-rfc6775-update-reqs].

The 6TiSCH Architecture extends the concepts of Deterministic Networking on a Layer-3 network. Work has started on this general problem with the DetNet Mailing lists and associated discussions. The 6TiSCH Architecture should inherit from that work and thus depends on it. In turn, DetNet must integrate and maintain consistency with the work that has taken place and is continuing at IEEE802.1TSN and AVnu.

The current charter positions 6TiSCH on IEEE802.15.4 only. Though most of the design should be portable on other link types, 6TiSCH has a strong dependency on IEEE802.15.4 and its evolution. A new version of the IEEE802.15.4 standard is expected in 2015. That version should integrate TSCH as well as other amendments and fixes into the main specification. The impact on this Architecture should be minimal to non-existent, but deeper work such as 6top and security may be impacted. A 6TiSCH Interest Group was formed at IEEE to maintain the synchronization and help foster work at the IEEE should 6TiSCH demand it.

ISA100 [ISA100] Common Network Management (CNM) is another external work of interest for 6TiSCH. The group, referred to as ISA100.20, defines a Common Network Management framework that should enable the management of resources that are controlled by heterogeneous protocols such as ISA100.11a [ISA100.11a], WirelessHART [WirelessHART], and 6TiSCH. Interestingly, the establishment of 6TiSCH Deterministic paths, called tracks, are also in scope, and ISA100.20 is working on requirements for DetNet.

6. 6LoWPAN (and RPL)

This architecture expects that a 6LoWPAN node can connect as a leaf to a RPL network, where the leaf support is the minimal functionality to connect as a host to a RPL network without the need to participate to the full routing protocol. The support of leaf can be implemented as a minor increment to 6LoWPAN ND, with the additional capability to carry a sequence number that is used to track the movements of the device, and optionally some information about the RPL topology that this device will join.

The root of the RPL network is integrated with the 6LoWPAN ND 6LBR, but it is logically separated from the 6BBR that is used to connect the RPL topology to the backbone. The RPL root can use Efficient ND as the interface to register an LLN node in its topology to the 6BBR for whatever operation the 6BBR performs, such as ND proxy operations, or injection in a routing protocol. It results that, as illustrated in Figure 4, the periodic signaling could start at the leaf node with 6LoWPAN ND, then would be carried over RPL to the RPL root, and then with Efficient-ND to the 6BBR. Efficient ND being an adaptation of 6LoWPAN ND, it makes sense to keep those two homogeneous in the way they use the source and the target addresses in the Neighbor Solicitation (NS) messages for registration, as well as in the options that they use for that process.

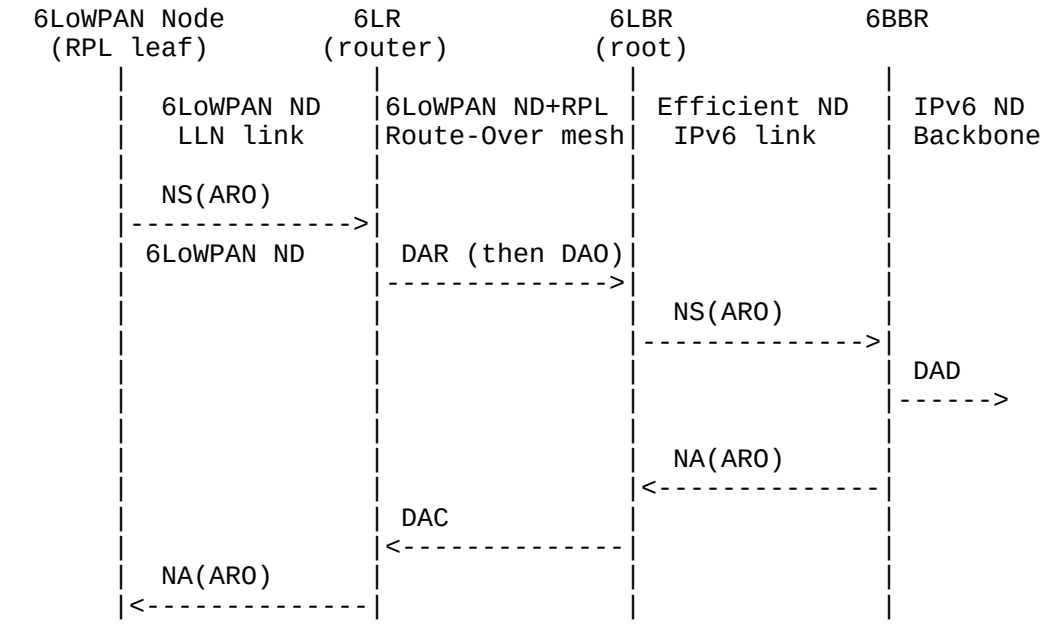


Figure 4: (Re-)Registration Flow over Multi-Link Subnet

As the network builds up, a node should start as a leaf to join the RPL network, and may later turn into both a RPL-capable router and a 6LR, so as to accept leaf nodes to recursively join the network.

6.1. RPL Leaf Support in 6LoWPAN ND

RPL needs a set of information in order to advertise a leaf node through a DAO message and establish reachability.

At the bare minimum the leaf device must provide a sequence number that matches the RPL specification in section 7. Section 4.1 of [I-D.chakrabarti-nordmark-6man-efficient-nd], on the Address Registration Option (ARO), already incorporates that addition with a new field in the option called the Transaction ID.

If for some reason the node is aware of RPL topologies, then providing the RPL InstanceID for the instances to which the node wishes to participate would be a welcome addition. In the absence of such information, the RPL router must infer the proper instanceID from external rules and policies.

On the backbone, the InstanceID is expected to be mapped onto a VLANID. Neither WiFi nor Efficient ND do provide a mapping to VLANIDs, and it is unclear, when a wireless node attaches to a backbone where VLANs are defined, which VLAN the wireless device attaches to. Considering that a VLAN is effectively the IP link on the backbone, adding the InstanceID to both specifications could be a welcome addition.

6.2. registration Failures Due to Movement

Registration to the 6LBR through DAR/DAC messages [RFC6775] may percolate slowly through an LLN mesh, and it might happen that in the meantime, the 6LoWPAN node moves and registers somewhere else. Both RPL and 6LoWPAN ND lack the capability to indicate that the same node is registered elsewhere, so as to invalidate states down the deprecated path.

In its current expression and functionality, 6LoWPAN ND considers that the registration is used for the purpose of DAD only as opposed to that of achieving reachability, and as long as the same node registers the IPv6 address, the protocol is functional. In order to act as a RPL leaf registration protocol and achieve reachability, the device must use the same TID for all its concurrent registrations, and registrations with a past TID should be declined. The state for an obsolete registration in the 6LR, as well as the RPL routers on the way, should be invalidated. This can only be achieved with the addition of a new Status in the DAC message, and a new error/clean-up flow in RPL.

6.3. Proxy registration

The 6BBR provides the capability to defend an address that is owned by a 6LoWPAN Node, and attract packets to that address, whether it is done by proxying ND over a MultiLink Subnet, redistributing the address in a routing protocol or advertising it through an alternate proxy registration such as the Locator/ID Separation Protocol [RFC6830] (LISP) or Mobility Support in IPv6 [RFC6275] (MIPv6). In a LLN, it makes sense to piggyback the request to proxy/defend an address with its registration.

6.4. Target Registration

In their current incarnations, both 6LoWPAN ND and Efficient ND expect that the address being registered is the source of the NS(ARO) message and thus impose that a Source Link-Layer Address (SLLA) option be present in the message. In a mesh scenario where the 6LBR is physically separated from the 6LoWPAN Node, the 6LBR does not own the address being registered. This suggests that [I-D.chakrabarti-nordmark-6man-efficient-nd] should evolve to register the Target of the NS message as opposed to the Source Address. From another perspective, it may happen, in the use case of a Star topology, that the 6LR, 6LBR and 6BBR are effectively collapsed and should support 6LoWPAN ND clients. The convergence of efficient ND and 6LoWPAN ND into a single protocol is thus highly desirable.

In any case, as long as the DAD process is not complete for the address used as source of the packet, it is against the current practice to advertise the SLLA, since this may corrupt the ND cache of the destination node, as discussed in the Optimistic DAD specification [RFC4429] with regards to the TENTATIVE state.

This may look like a chicken and an egg problem, but in fact 6LoWPAN ND acknowledges that the Link-Local Address that is based on an EUI-64 address of a LLN node may be autoconfigured without the need for DAD. It results that a node could use that Address as source, with an SLLA option in the message if required, to register any other addresses, either Global or Unique-Local Addresses, which would be indicated in the Target.

The suggested change is to register the target of the NS message, and use Target Link-Layer Address (TLLA) in the NS as opposed to the SLLA in order to install a Neighbor Cache Entry. This would apply to both Efficient ND and 6LoWPAN ND in a very same manner, with the caveat that depending on the nature of the link between the 6LBR and the 6BBR, the 6LBR may resort to classical ND or DHCPv6 to obtain the

address that it uses to source the NS registration messages, whether for itself or on behalf of LLN nodes.

6.5. RPL root vs. 6LBR

6LoWPAN ND is unclear on how the 6LBR is discovered, and how the liveliness of the 6LBR is asserted over time. On the other hand, the discovery and liveliness of the RPL root are obtained through the RPL protocol.

When 6LoWPAN ND is coupled with RPL, it makes sense to collocate the 6LBR and the RPL root functionalities. The DAR/DAC exchange becomes a preamble to the DAO messages that are used from then on to reconfirm the registration, thus eliminating a duplication of functionality between DAO and DAR messages.

6.6. Securing the Registration

A typical attack against IPv6 ND is address spoofing, whereby a rogue node claims the IPv6 Address of another node in and hijacks its traffic.

SEcure Neighbor Discovery (SEND) [RFC3971] is designed to protect each individual ND lookup/advertisement in a peer to peer model where each lookup may be between different parties. This is not the case in a 6LoWPAN ND LLN where, as illustrated in Figure 4, the 6LBR terminates all the flows and may store security information for later validation.

Additionally SEND requires considerably enlarged ND messages to carry cryptographic material, and requires that each protected address is generated cryptographically, which implies the computation of a different key for each Cryptographically Generated Address (CGA). SEND as defined in [RFC3971] is thus largely unsuitable for application in a LLN.

Once an Address is registered, the 6LBR maintains a state for that Address and is in position to bind securely the first registration with the Node that placed it, whether the Address is CGA or not. It should thus be possible to protect the ownership of all the addresses of a 6LoWPAN Node with a single key, and there should not be a need to carry the cryptographic material more than once to the 6LBR.

The energy constraint is usually a foremost factor, and attention should be paid to minimize the burden on the CPU. Hardware-assisted support of variants of the Counter with CBC-MAC [RFC3610] (CCM) authenticated encryption block cipher mode such as CCM* are common in

LowPower ship-set implementations, and 6LoWPAN ND security mechanism should be capable to reuse them when applicable.

Finally, the code footprint in the device being also an issue, the capability to reuse not only hardware-assist mechanisms but also software across layers has to be considered. For instance, if code has to be present for upper-layer operations, e.g AES-CCM Cipher Suites for Transport Layer Security (TLS) [RFC6655], then the capability to reuse that code should be considered.

7. 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.

section 2.1.3 of [I-D.ietf-roll-rpl-industrial-applicability] and next sections discuss application-layer paradigms, such as Source-sink (SS) that is a Multipeer to Multipeer (MP2MP) model 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 9. Link-layer frame forwarding interactions are discussed in Section 10, and Network-layer Packet routing is addressed in Section 11.

8. TSCH and 6top

8.1. 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 9.2.

The 6top data model and management interfaces are further discussed in Section 9.3.

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 devices communicating over this cell negotiate its new position in the TSCH schedule.

8.2. 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 (e.g. RSSI or 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 for instance

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 protocol exchanges by an upper layer such as 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.

8.3. 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 acknowledgment-based synchronization. In order to maximize battery life and network throughput, it is advisable that RPL ICMP discovery and maintenance traffic (governed by the trickle timer) be somehow coordinated with the transmission of time 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 section 3.1.3 of RPL [RFC6550], "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 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 section 3.5.1 of [RFC6550], "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.

8.4. 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) and a width (in timeSlots) that is the period of the network scheduling operation (indexed by slotOffsets) for that CDU matrix. The size of a cell is a timeSlot duration, and values of 10 to 15 milliseconds are typical in 802.15.4e TSCH to accommodate for the transmission of a frame and an ack, including the security validation on the receive side which may take up to a few milliseconds on some device architecture.

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 reduce 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.

When a packet is received from a higher layer for transmission, 6top inserts that packet in the outgoing queue which matches the packet best (Differentiated Services [RFC2474] 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.

8.5. 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 fulfill 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. Note that a PCE is expected to have precedence in the allocation, so that a RPL parent would only be able to obtain portions that are not in-use by the PCE.

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 chunk is the basic unit of ownership that is used in that process.

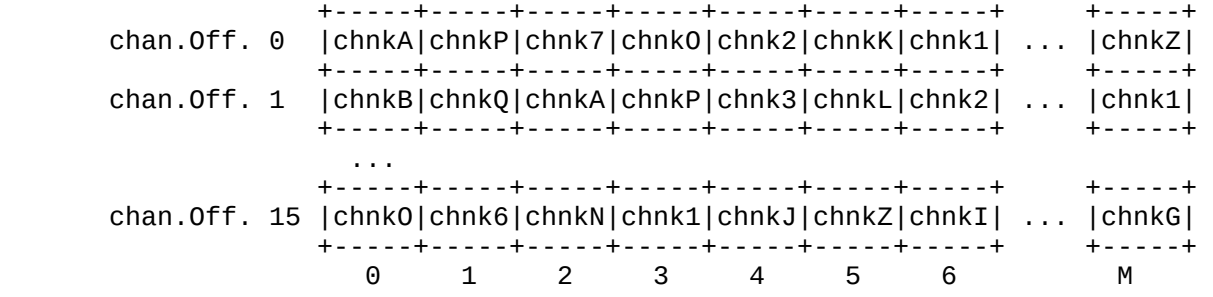


Figure 5: CDU matrix Partitioning in Chunks

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.

9. 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.

9.1. 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 schedule 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].

9.2. Neighbor-to-neighbor Scheduling

In the simplest instantiation of a 6TiSCH network described in Section 9.1, 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 reallocation 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.

9.3. 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 of 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.

The capability to interact with the node 6top sublayer from multiple hops away can be leveraged for monitoring, scheduling, or a combination of thereof. The architecture supports variations on the deployment model, and focuses on the flows rather than whether there is a proxy or a translation operation en-route.

[I-D.ietf-6tisch-coap] defines an mapping of the 6top set of commands, which is 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).

At the time of this writing, a Deterministic Networking (DetNet) [I-D.finn-detnet-problem-statement] effort has started at the IETF to provide homogeneous flows and services across layers. This architecture will be refined to comply with DetNet when the work is formalized.

9.4. 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 expected to be similar in essence to [RFC3209] and/or [RFC4080]/[RFC5974]. The protocol for a node to trigger hop-by-hop scheduling is not yet defined.

10. Forwarding Models

By forwarding, this specification means the per-packet operation that allows to deliver a packet to a next hop or an upper layer in this node. Forwarding is based on pre-existing state that was installed as a result of a routing computation Section 11. 6TiSCH supports three different forwarding model, G-MPLS Track Forwarding (TF), 6LoWPAN Fragment Forwarding (FF) and IPv6 Forwarding (6F).

10.1. Track Forwarding

A Track is a unidirectional path between a source and a destination. In a Track cell, the normal operation of IEEE802.15.4e Automatic Repeat-reQuest (ARQ) usually happens, though the acknowledgment may be omitted in some cases, for instance if there is no scheduled cell for a retry.

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).

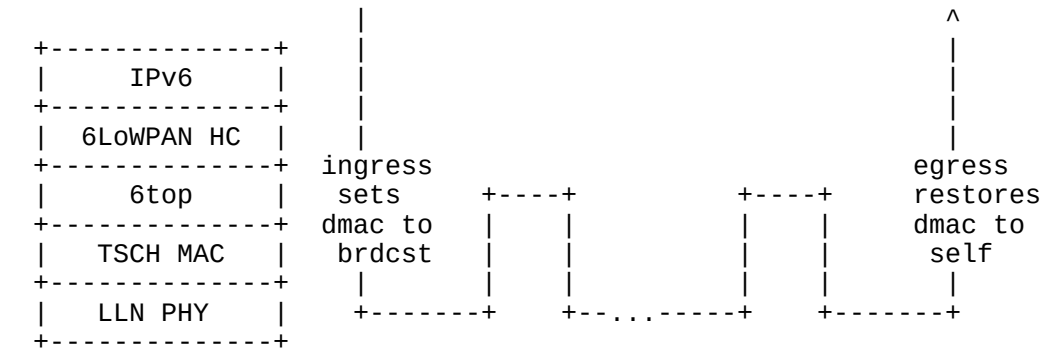
On the other hand, it might happen that there are not enough TX-cells in the transmit bundle to accommodate the Track traffic, for instance if more retransmissions are needed than provisioned. In that case, the frame can be placed for transmission in the bundle that is used for layer-3 traffic towards 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 11.1. 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.

10.1.1. 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

10.1.2. 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.

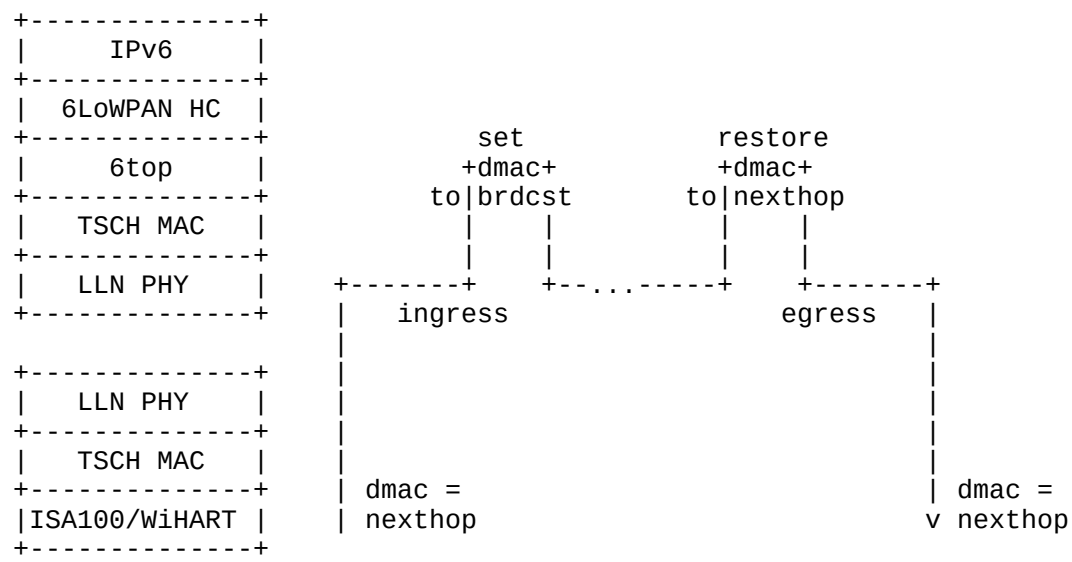


Figure 6: Track Forwarding, Tunnel Mode

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 tunneled 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.

10.1.3. 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.

10.2. 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 based on an Explicit Congestion Notification, as well as end-to-end individual fragment recovery.

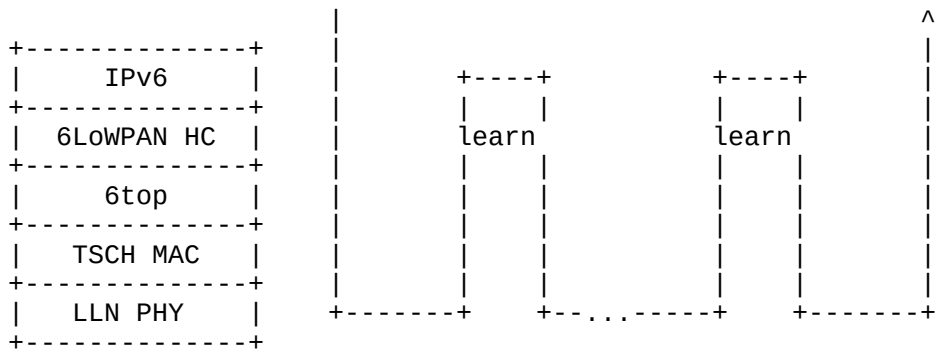


Figure 7: Forwarding First Fragment

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.

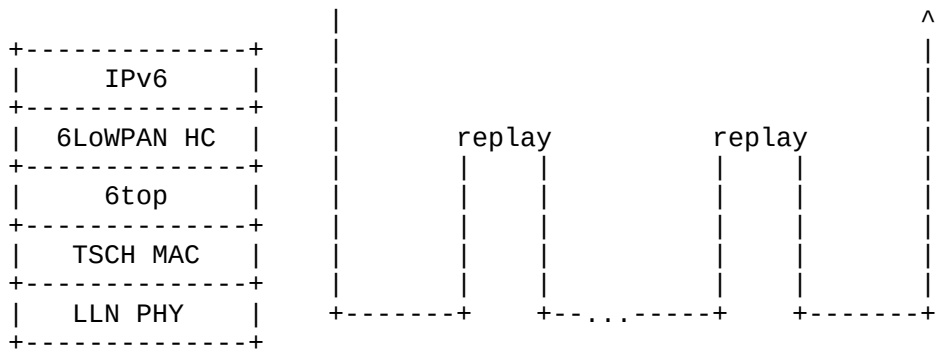


Figure 8: Forwarding Next Fragment

A bitmap and an ECN echo in the end-to-end acknowledgment 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.

10.3. 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].

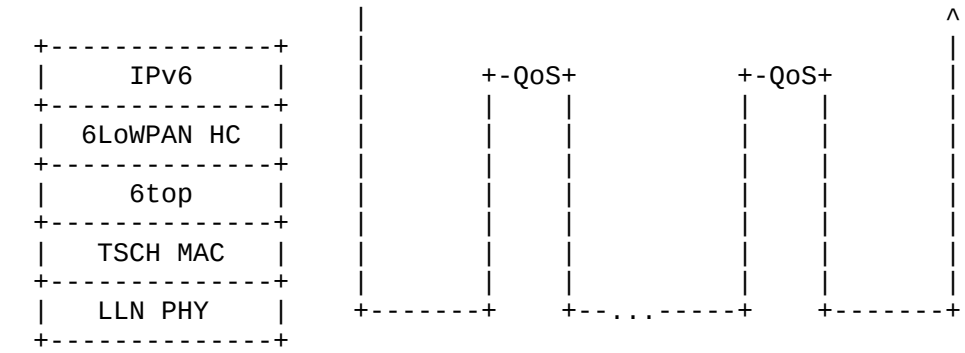


Figure 9: IP Forwarding

11. Centralized vs. Distributed Routing

6TiSCH supports a mixed model of centralized routes and distributed routes. Centralized routes can for example be computed by a 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 a 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.

11.1. 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, that information is the RPLInstanceID which is carried in the RPL Packet Information, as discussed in section 11.2 of [RFC6550], "Loop Avoidance and Detection".

The RPL Packet Information (RPI) is carried in IPv6 packets as a RPL option in the IPv6 Hop-By-Hop Header [RFC6553].

6Lo is currently considering a Next Header Compression (NHC) for the RPI (RPI-NHC). The RPI-NHC is specified in [I-D.thubert-6lo-rpl-nhc], and is the compressed equivalent to the whole HbH header with the RPL option.

An alternative form of compression that integrates the compression on IP-in-IP encapsulation and the Routing Header type 3 [RFC6554] with that of the RPI in a new 6LoWPAN dispatch/header type is concurrently being evaluated as [I-D.thubert-6lo-routing-dispatch].

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.

12. IANA Considerations

This specification does not require IANA action.

13. Security Considerations

This architecture operates on IEEE802.15.4 and expects link-layer security to be enabled at all times between connected devices, except for the very first step of the device join process, where a joining device may need some initial, unsecured exchanges so as to obtain its initial key material. Work has already started at the 6TiSCH Security Design Team and an overview of the current state of that work is presented in Section 13.1.

Future work on 6TiSCH security and will examine in deeper detail how to secure transactions end-to-end, and to maintain the security posture of a device over its lifetime. The result of that work will be described in a subsequent volume of this architecture.

13.1. Join Process Highlights

The architecture specifies three logical elements to describe the join process:

A Joining Node (JN): Node that wishes to become part of the network;

A Join Coordination Entity (JCE) : A Join Coordination Entity (JCE) that arbitrates network access and hands out network parameters (such as keying material);

A Join Assistant (JA), a one-hop (radio) neighbor of the joining node that acts as proxy network node and may provide connectivity with the JCE.

The join protocol consists of three phases:

Device Authentication: The JN and the JA mutually authenticate each other and establish a shared key, so as to ensure on-going authenticated communications. This may involve a server as a third party.

Authorization: The JA decides on whether/how to authorize a JN (if denied, this may result in loss of bandwidth). Conversely, the JN decides on whether/how to authorize the network (if denied, it will not join the network). Authorization decisions may involve other nodes in the network.

Configuration/Parameterization: The JA distributes configuration information to the JN, such as scheduling information, IP address assignment information, and network policies. This may originate from other network devices, for which the JA may act as proxy. This step may also include distribution of information from the JN to the JA and other nodes in the network and, more generally, synchronization of information between these entities.

The device joining process is depicted in Figure 10, where it is assumed that devices have access to certificates and where entities have access to the root CA keys of their communicating parties (initial set-up requirement). Under these assumptions, the authentication step of the device joining process does not require online involvement of a third party. Mutual authentication is performed between the JN and the JA using their certificates, which also results in a shared key between these two entities.

The JA assists the JN in mutual authentication with a remote server node (primarily via provision of a communication path with the

server), which also results in a shared (end-to-end) key between those two entities. The server node may be a JCE that arbitrages the network authorization of the JN (where the JA will deny bandwidth if authorization is not successful); it may distribute network-specific configuration parameters (including network-wide keys) to the JN. In its turn, the JN may distribute and synchronize information (including, e.g., network statistics) to the server node and, if so desired, also to the JA. The actual decision of the JN to become part of the network may depend on authorization of the network itself.

The server functionality is a role which may be implemented with one (centralized) or multiple devices (distributed). In either case, mutual authentication is established with each physical server entity with which a role is implemented.

Note that in the above description, the JA does not solely act as a relay node, thereby allowing it to first filter traffic to be relayed based on cryptographic authentication criteria - this provides first-level access control and mitigates certain types of denial-of-service attacks on the network at large.

Depending on more detailed insight in cost/benefit trade-offs, this process might be complemented by a more "relaxed" mechanism, where the JA acts as a relay node only. The final architecture will provide mechanisms to also cover cases where the initial set-up requirements are not met or where some other out-of-sync behavior occurs; it will also suggest some optimizations in case JCE-related information is already available with the JA (via caching of information).

When a device rejoins the network in the same authorization domain, the authorization step could be omitted if the server distributes the authorization state for the device to the JA when the device initially joined the network. However, this generally still requires the exchange of updated configuration information, e.g., related to time schedules and bandwidth allocation.

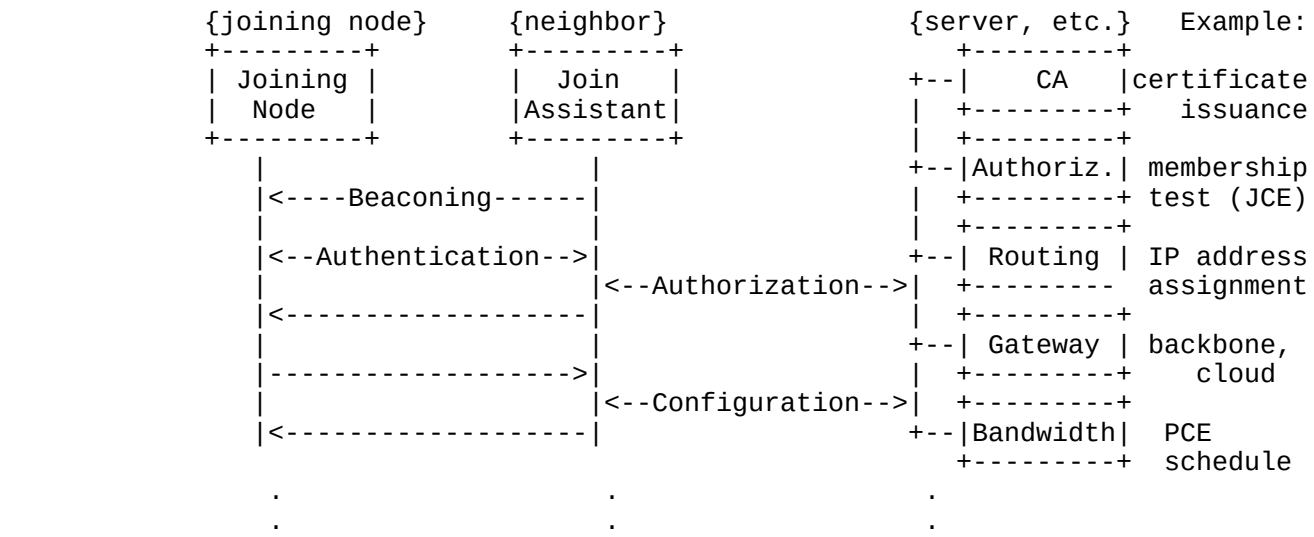


Figure 10: Network joining, with only authorization by third party

14. Acknowledgments

14.1. Contributors

The editors and authors wish to recognize the contribution of

Kris Pister for creating it all and his continuing guidance through the elaboration of this design.

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.

Robert Assimiti for his breakthrough work on RPL over TSCH and initial text and guidance.

14.2. Special Thanks

Special thanks to Tero Kivinen, Jonathan Simon, Giuseppe Piro, Subir Das and Yoshihiro Ohba for their deep contribution to the initial security work, and to Diego Dujovne for starting and leading the On-the-Fly effort.

Special thanks also to Pat Kinney for his support in maintaining the connection active and the design in line with work happening at IEEE802.15.4.

Also special thanks to Ted Lemon who was the INT Area A-D while this specification was developed for his great support and help throughout.

14.3. And Do not Forget

This specification is the result of multiple interactions, in particular during the 6TiSCH (bi)Weekly Interim call, relayed through the 6TiSCH mailing list at the IETF.

The authors wish to thank: Alaeddine Weslati, Chonggang Wang, Georgios Exarchakos, Zhuo Chen, Alfredo Grieco, Bert Greevenbosch, Cedric Adjih, Deji Chen, Martin Turon, Dominique Barthel, Elvis Vogli, Geraldine Texier, Malisa Vucinic, Guillaume Gaillard, Herman Storey, Kazushi Muraoka, Ken Bannister, Kuor Hsin Chang, Laurent Toutain, Maik Seewald, Maria Rita Palattella, Michael Behringer, Nancy Cam Winget, Nicola Accettura, Nicolas Montavont, Oleg Hahm, Patrick Wetterwald, Paul Duffy, Peter van der Stock, Rahul Sen, Pieter de Mil, Pouria Zand, Rouhollah Nabati, Rafa Marin-Lopez, Raghuram Sudhaakar, Sedat Gormus, Shitanshu Shah, Steve Simlo, Tengfei Chang, Tina Tsou, Tom Phinney, Xavier Lagrange, Ines Robles and Samita Chakrabarti for their participation and various contributions.

15. References

15.1. Normative 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-03 (work in progress), January 2015.

[I-D.ietf-6tisch-tsch]
Watteyne, T., Palattella, M., and L. Grieco, "Using IEEE802.15.4e TSCH in an IoT context: Overview, Problem Statement and Goals", draft-ietf-6tisch-tsch-05 (work in progress), January 2015.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[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.

[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.

[RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, 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.

15.2. 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-07 (work in progress), February 2015.

[I-D.dujovne-6tisch-on-the-fly]
Dujovne, D., Grieco, L., Palattella, M., and N. Accettura, "6TiSCH On-the-Fly Scheduling", draft-dujovne-6tisch-on-the-fly-04 (work in progress), January 2015.

[I-D.finn-detnet-problem-statement]
Finn, N. and P. Thubert, "Deterministic Networking Problem Statement", draft-finn-detnet-problem-statement-01 (work in progress), October 2014.

[I-D.ietf-6tisch-6top-interface]
Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top) Interface", draft-ietf-6tisch-6top-interface-02 (work in progress), October 2014.

[I-D.ietf-6tisch-coap]
Sudhaakar, R. and P. Zand, "6TiSCH Resource Management and Interaction using CoAP", draft-ietf-6tisch-coap-02 (work in progress), December 2014.

[I-D.ietf-6tisch-minimal]
Vilajosana, X. and K. Pister, "Minimal 6TiSCH Configuration", draft-ietf-6tisch-minimal-05 (work in progress), January 2015.

[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.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.svshah-tsvwg-deterministic-forwarding]
Shah, S. and P. Thubert, "Deterministic Forwarding PHB", draft-svshah-tsvwg-deterministic-forwarding-03 (work in progress), March 2015.

[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-04 (work in progress), February 2015.

[I-D.thubert-6lo-rfc6775-update-reqs]
Thubert, P. and P. Stok, "Requirements for an update to 6LoWPAN ND", draft-thubert-6lo-rfc6775-update-reqs-06 (work in progress), January 2015.

[I-D.thubert-6lo-routing-dispatch]
Thubert, P., Bormann, C., Toutain, L., and R. Cragie, "A Routing Header Dispatch for 6LoWPAN", draft-thubert-6lo-routing-dispatch-03 (work in progress), January 2015.

[I-D.thubert-6lo-rpl-nhc]
Thubert, P. and C. Bormann, "A compression mechanism for the RPL option", draft-thubert-6lo-rpl-nhc-02 (work in progress), October 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-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.vanderstok-core-comi]
Stok, P., Greevenbosch, B., Bierman, A., Schoenwaelder, J., and A. Sehgal, "CoAP Management Interface", draft-vanderstok-core-comi-06 (work in progress), February 2015.

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

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[RFC2545] Marques, P. and F. Dupont, "Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing", RFC 2545, March 1999.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

[RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", RFC 3444, January 2003.

[RFC3610] Whiting, D., Housley, R., and N. Ferguson, "Counter with CBC-MAC (CCM)", RFC 3610, September 2003.

[RFC3963] Devarapalli, V., Wakikawa, R., Petrescu, A., and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol", RFC 3963, January 2005.

[RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

[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.

[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.

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, July 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.

[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.

[RFC6655] McGrew, D. and D. Bailey, "AES-CCM Cipher Suites for Transport Layer Security (TLS)", RFC 6655, July 2012.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, January 2013.

15.3. Other Informative References

[CCAMP] IETF, "Common Control and Measurement Plane", <https://datatracker.ietf.org/doc/charter-ietf-ccamp/>.

[DICE] IETF, "DTLS In Constrained Environments", <https://datatracker.ietf.org/doc/charter-ietf-dice/>.

[HART] www.hartcomm.org, "Highway Addressable remote Transducer, a group of specifications for industrial process and control devices administered by the HART Foundation", .

[IEEE802.1TSNTG] IEEE Standards Association, "IEEE 802.1 Time-Sensitive Networks Task Group", March 2013, <http://www.ieee802.org/1/pages/avbridges.html>.

[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] ISA/ANSI, "ISA100, Wireless Systems for Automation", <https://www.isa.org/isa100/>.

[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>.

[PCE] IETF, "Path Computation Element", <https://datatracker.ietf.org/doc/charter-ietf-pce/>.

[TEAS] IETF, "Traffic Engineering Architecture and Signaling", <https://datatracker.ietf.org/doc/charter-ietf-teas/>.

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

Appendix A. Personal submissions relevant to the next volumes

This volume only covers a portion of the total work that is needed to cover the full 6TiSCH architecture. Missing portions include Deterministic Networking with Track Forwarding, Dynamic Scheduling, and Security.

[I-D.richardson-6tisch-security-architecture] elaborates on the potential use of 802.1AR certificates, and some options for the join process are presented in more details.

[I-D.dujovne-6tisch-on-the-fly] discusses the use of the 6top sublayer [I-D.wang-6tisch-6top-sublayer] to adapt dynamically the number of cells between a RPL parent and a child to the needs of the actual traffic.

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

Rene Struik
Struik Security Consultancy

Email: rstruik.ext@gmail.com

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

Email: mcr+ietf@sandelman.ca
URI: http://www.sandelman.ca/

6TiSCH
Internet-Draft
Intended status: Standards Track
Expires: September 10, 2015

R. Sudhaakar, Ed.
Cisco
P. Zand
University of Twente
March 9, 2015

6TiSCH Resource Management and Interaction using CoAP
draft-ietf-6tisch-coap-03

Abstract

The [IEEE802154e] standardizes the TSCH mode of operation and defines the mechanisms for layer 2 communication between conforming devices. 6top defines a set of commands to monitor and manage the TSCH schedule. To realize the full functionality of sensor networks and allow their adoption and use in real applications we need additional mechanisms. Specifically, the interaction with 6top, control and modify schedules, monitor parameters etc must be defined. Higher layers monitoring and management entities are then able to use these capabilities to create feedback loops. Although, there have been many custom implementations of such feedback loops between the routing, transport and MAC layers in sensor network deployments, there has been a lack of standards based approaches. This draft defines the messaging between monitoring and management entities and the 6top layer and a mapping to the 6top commands. The document also presents a particular implementation of the generic data model specified in [I-D.ietf-6tisch-6top-interface] based on CoAP and CBOR.

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 September 10, 2015.

Table of Contents

1. Requirements notation

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

2. Introduction

The 6TiSCH Operation Sublayer (6top) [I-D.ietf-6tisch-6top-interface] describes the main commands provided to higher layers that allow them to build TSCH schedules, make routing decisions, perform TSCH configuration and control procedures and supports centralized and decentralized scheduling policies among other functionalities. However, there is still a need for specifying the methods, including message exchanges and message formats that higher layers use to invoke these command described by 6top.

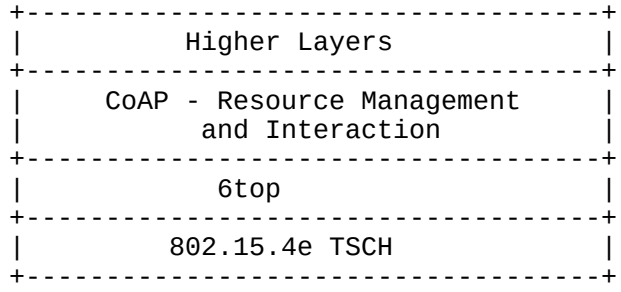


Figure 1: Logical positioning of layers

Interoperation with any protocol that may be used by the network layer is necessary to have a wide impact. This documents aims at defining the message exchanges and the formats of the messages that the network layer uses to interact with the 6top sub-layer. The messaging scheme defined in this document is aimed for use between 6top nodes and higher layer management entities as well as between 6top nodes.

This document also specifies an implementation of this generic message exchange and data model using CoAP as the transport mechanism.

3. Scope of the document

This draft defines the communication mechanism between PCE and 6top nodes using COAP. The generic YANG data model defined in [I-D.ietf-6tisch-6top-interface] is used to define the various CoAP messages and payloads. The payload used CBOR for the encoding

format. The document also defines the URIs that used to identify the resources exposed by 6top.

This document also defines how users can install custom resources that allow them to extend the basic resource exposed by 6top.

The CoAP Management Interface (CoMI) [I-D.vanderstok-core-comi] draft specifies a common constrained device managment interface. The conventions used in this draft follow the guidelines in the CoMI draft . This draft expects CoMI to define the access methodologies, discovery mechanisms, resource installation procedures required for the management of constrained devices. This draft presents some examples in Section 4.3.4 on how to use the CoMI specifications to manage the 6top sublayer.

NOTE: CoMI specifications are not finalized at the time of this writing. In case of any discrepancies, CoMI will supersede the message formats in the examples presented in Section 4.3.4.

4. Data Model definition for CoAP

4.1. Naming Convention for URI schemes

Universal Resource Identifiers (URIs) help us uniquely identify the various commands and parameters that 6top exposes to the higher layers. The basic URI naming conventions and terminology specified in [RFC3986] is used. Specifically, the terms, ’scheme’, ’authority’, ’path’, ’query’ are used as defined in the [RFC3986].

The following provides the guidelines that are followed in this draft to name the URIs that identify the resources exposed by 6top.

1. All URIs naming 6top resources MUST use the ’coap’ scheme

2. The authority MUST have the username ’6top’ and the IP address of 6top node

3. The root path MUST always start with ’6top’

4. Each component of the path SHOULD be of minimum possible length while being self descriptive.

5. Typographical conventions as described in A SHOULD be followed

These guidelines MUST be followed by users who install extensible resources. It SHOULD be followed for future extensions of the data model in order to provide consistency.

4.2. Convention for accessing URIs

We use the GET, POST and DELETE methods described by CoAP. These methods MUST be used in accordance with their definition in Sec. 5.8 of [RFC7252] and as specified in the CoMI draft [I-D.vanderstok-core-comi]. There is no need for the PUT method as the functionality of the POST method can be used for all situations that need updating or modification of a resource.

The CoAP methods are mapped to 6top commands as shown in the figure below.

CoAP method	6top command	Description
GET	READ	Retrieves 6top resources
POST	CREATE / UPDATE	Creates/Updates a new entry
DELETE	DELETE	Deletes an entry
POST	CONFIGURE	Configures a setting

Figure 2: Mapping between CoAP methods and 6top commands

The GET method may use queries to allow higher layer entities to perform conditional GETs or filter the results of a GET on resource that is a collection.

The POST method is used in all situations where an argument needs to be passed to the 6top layer. The Content-Type option is set to 'application/cbor'. The payload is encoded using CBOR format as described in [I-D.vanderstok-core-comi] and [RFC7049].

The DELETE method is used to invoke the 6top DELETE command on a particular resource.

The GET method may use queries to allow higher layer entities to perform conditional GETs or filter the results of a GET on resource that is a collection.

4.3. 6TiSCH Resources

The 6TiSCH resources presented in this draft offer a comprehensive way to manage 6top nodes based on the requirement known to us as of this writing. These resources are bound to evolve and will be

identified by appropriate version numbers that will be tied to revisions of this draft.

Management resources are classified as resources to which a higher layer entity may create, update or delete. They are typically used to create schedules, identify time sources that TSCH needs. They are the means to close the control loop between TSCH and higher layers.

Informational resources are classified as resources to which a higher layer entity typically has only READ access. They are typically used to monitor operational parameters of TSCH and the values used as input to routing algorithms and other mechanisms.

4.3.1. Versioning

The version number describes the set of resources that can be accessed on a node that implements the recommendations in this draft.

Each revision of this draft will define a version number which will uniquely identify the set of resources defined in that particular revision of the draft. Specifically, a change to the major version number indiactes that resources have been added, deleted, renamed or their message formats have changed. In most cases, this MAY require changes to the implementation. The minor version number indicates changes to options supported by resources or other textual/language changes to the draft. In most cases, this MAY NOT require any changes to the implementation.

The 6TiSCH resource version information is available by executing a GET method on the resource ’/6top/version’ of a 6top node.

4.3.2. Management Resources

All the attributes in the management resources have the Read/Write accessibility. The following table lists the 6top management resources and the related URI paths.

Name	Accessibility 6top Commands	URI path
Neighbor List	CREATE/READ/ DELETE/UPDATE	6top/nbrList
slotframe List	CREATE/READ/ DELETE/UPDATE	6top/slotFrame
Cell List	CREATE/READ/ DELETE/UPDATE	6top/cellList
Time Source	CREATE/READ/ DELETE/UPDATE	6top/timeSource
LabelSwitch List	CREATE/READ/ DELETE/UPDATE	6top/labelSwitch
Track List	CREATE/READ/ DELETE/UPDATE	6top/tracklist
EB List	CREATE/READ/ DELETE/UPDATE	6top/ebList
Chunk List	CREATE/READ/ DELETE/UPDATE	6top/chunkList

Figure 3: List of Management Resources

The following table provides an example about how Neighbor List components (leafs in the YANG model) can be addressed.

Field name	URI path
Target Neighbor Addr	6top/nbrList/tna
ASN	6top/nbrList/asn
RSSI	6top/nbrList/rssi
LinkQuality	6top/nbrList/linkQ

Figure 4: Neighbor Table

4.3.3. Informational Resources

All the attributes in the Informational resources have the Read accessibility. The following table lists the 6top informational resources and the related URI paths.

Name	Accessibility 6top Commands	URI path
Version	READ	6top/version
Queue	READ/CONFIGURE	6top/queue
Monitoring status	READ/CONFIGURE	6top/monitStatus
Statistics metrics	READ/CONFIGURE	6top/stats

Figure 5: List of Informational Resources

4.3.3.1. Version

The version resource is a read-only resource that provides information on the methods, resources, message formats that is supported by the node. The version resource does not directly map to a 6top resource defined in [I-D.ietf-6tisch-6top-interface].

Upon receiving a GET on the '/6top/version' resource, the node MUST respond with a version number that is described by a major and minor

number. It is expressed using 2 bytes - The first and second bytes are 8-bit unsigned integers indicating the major and minor version numbers respectively. The valid values for the major version are 1 through 255 (both inclusive) and for the minor version are 0 through 255 (both inclusive).

A 6top node implmenting the recommendations in this draft will respond with the following 2 byte version number - '0x01 0x00', indicating major version = 1 and minor version = 0.

The major and minor versions are separately accessible using the resources '/6top/version/major' and '/6top/version/minor' respectively. The response will be an 8-bit unsigned integer containing the major or minor version number, respectively.

4.3.3.2. Resource Discovery

As new resources are defined (both native and custom), it is essential for the PCE as well peers to discover the resources. The CoMI draft presents methods by which standard CoAP resource discovery mechanisms are extended to the management of constrained devices. The methods described in Section 4.3 of [I-D.vanderstok-core-comi] SHALL be used for discovering new resources available at a node.

4.3.4. Message Formats

NOTE: The message formats presented in this section follow the specifications in the CoMI draft [I-D.vanderstok-core-comi]. In case of any discrepancies, the CoMI draft will take precedence.

GET messages do not contain any payload. However, they can contain a query option to filter on the resource that is being retrieved. An example query on the neighbor list is:

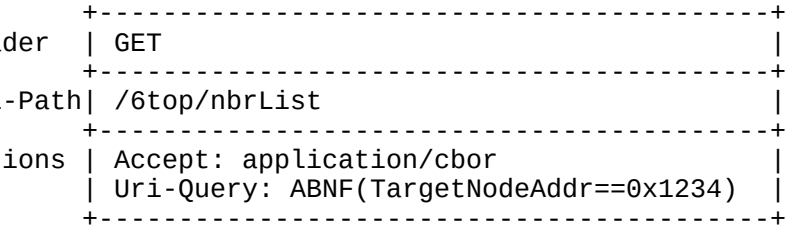


Figure 6: Example GET message

Since this resources points to the entire neighbor list, the response returns all the entries (the list of neighbors of that node) and all fields in each entry (i.e. entry for a neighbor) of the list in CBOR

format. A request with a Uri-Query option may be used to retrieve only specific entries in the list. The value of Uri-Query MUST be in the ABNF format as described in [RFC5234].

Resources that point to collection within a list, such as '/6top/nbrList/tna', returns only the values in the TargetNodeAddr entry of the Neighbor list. The usage of the Uri-Query option has the same effect of filtering on the result.

The endpoint MUST appropriately respond with a 2.05 Content or 4.04 Not Found message as defined in [RFC7252]. If the resource is found then the payload of the response MUST contain a CBOR representation of the data that is referenced by the URI.

To create or update a Neighbor, the CoAP client MUST send a POST message as shown in Figure 7. The payload MUST describe the argument that is passed to 6top in CBOR format.

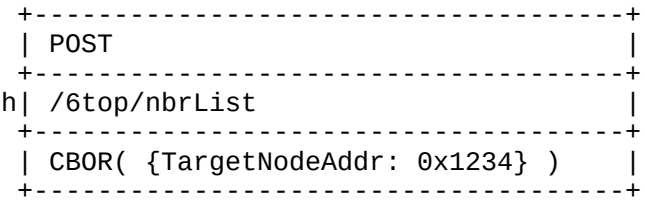


Figure 7: Example POST message

The POST method may not be used on resources that are collection within a list, such as '/6top/nbrList/tna'.

To delete a Neighbor, the CoAP client MUST send a DELETE message as shown in Figure 8.

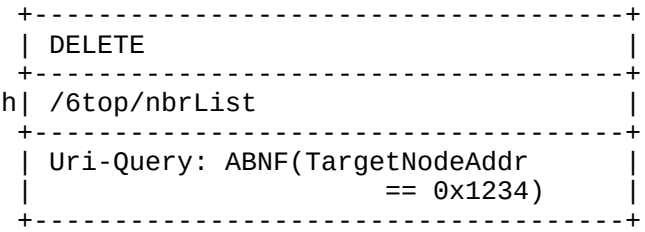


Figure 8: Example DELETE message

A DELETE message SHOULD always contain a Uri-Query option in order to clearly specify which entry(s) within the list must be deleted. Ideally, the CoAP client SHOULD make one call per entry that must be

deleted. An implementation may decide whether or not a DELETE method on '/6top/nbrList' may be allowed.

The endpoint MUST appropriately respond with a 2.02 (Deleted) message.

A sample of mapping between CoAP methods and 6top commands for manipulating the neighbor list is shown in the figure below.

CoAP method	6top command	6top behaviour	CoAP Response
POST /6top/nbrList CBOR({TargetNodeAddr: 1234})	Create.neighbor (address,stats)	Adds a neighbor	2.01 Created
GET /6top/nbrList	Read.all. neighbor()	Reads all neighbors	2.05 Content CBOR(Neighbor List)
GET /6top/nbrList Uri-Query - TargetNodeAddr: 1234})	Read.neighbor (address)	Reads neighbor information	2.05 Content CBOR(Neighbor List)
POST /6top/nbrList CBOR({TargetNodeAddr: 1234})	Update.neighbor (address,stats)	Updates an entry	2.04 Changed
DELETE /6top/nbrList Uri-Query - TargetNodeAddr == 1234})	Delete.neighbor (address)	Removes the neighbor	2.02 Deleted

Figure 9: CoAP methods and resulting invocation 6top commands

4.3.5. Extensible Resources

Extensible resources are to be used when a higher layer entity wants to be notified of an event. An event may be defined as the result of a mathematical operation on a 6top resource. For example, the CoAP client might want to monitor when the DAG rank of a particular node crosses a threshold. Once the extensible resource is installed the CoAP client uses the observe mechanism defined in [I-D.ietf-core-observe] to monitor the resource.

4.3.5.1. Defining new resources

An extensible resource path MUST always start with '/6top/custom' and follow the guideline for URI naming as described in 4.1. The event associated with the extensible resource must be defined using the ABNF notation described in [RFC5234].

An extensible resource may be created by performing POST operation to the resource '/6top/custom' with the following payload encoded using CBOR.

Field Name	Type
Resource Name	String
Event Definition	String

Figure 10: Payload format for creating an Extensible Resource

4.4. Example

This section gives a number of short examples of how to use the data model and CoAP mapping defined in this document.

4.4.1. Request-Response

Figure 11 shows how a CoAP client adds an entry in the neighbor list of node A. This new neighbor has a target node address 0x1234. The client sends out a POST request containing the CBOR encoding of '{TargetNodeAddr: 1234}'. This message is received and processed by the CoAP endpoint of Node A and in turn, the 6top command, Create.neighbor is invoked with the appropriate parameters. In this case, the address is the 'TargetNodeAddr' parameter passed in the payload of the POST message and the stats argument has the default value. In the response to the invocation of the Create.neighbor command, the 6top sublayer adds an entry to the neighbor list with appropriate values and returns a confirm message. The CoAP endpoint in turn send out an appropriate CoAP response to indicate success. If the addition of the neighbor failed, a failure message will be returned.

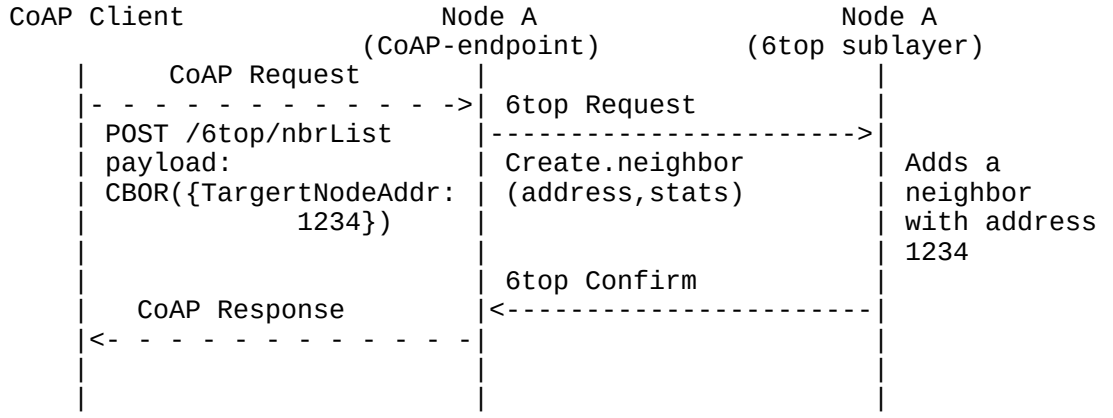


Figure 11: Example of adding a neighbor

In Figure 12, a CoAP client reads a neighbor entry from node A. This neighbor has a target node address 0x1234.

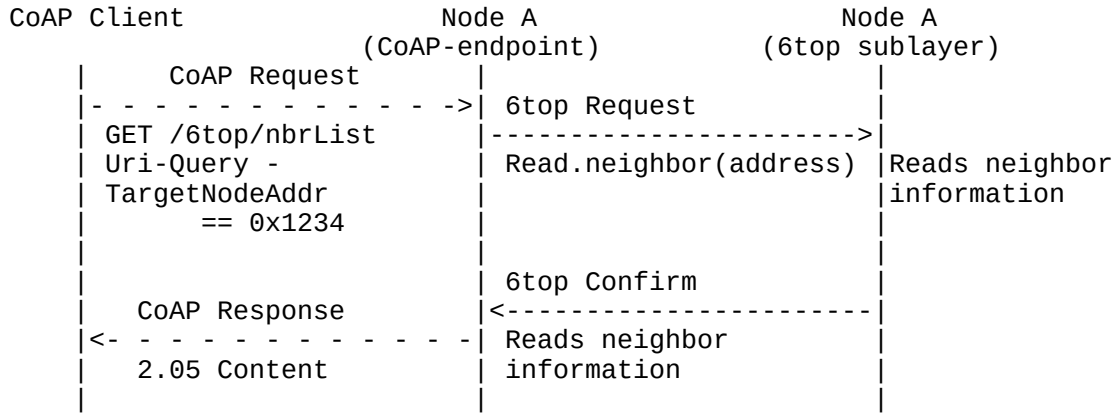


Figure 12: Example of reading a neighbor

4.4.2. Publish-Subscribe

In Figure 13, a CoAP client subscribes to Monitoring Status of node A. The Monitoring status of Node A is constantly monitored by the CoAP client.

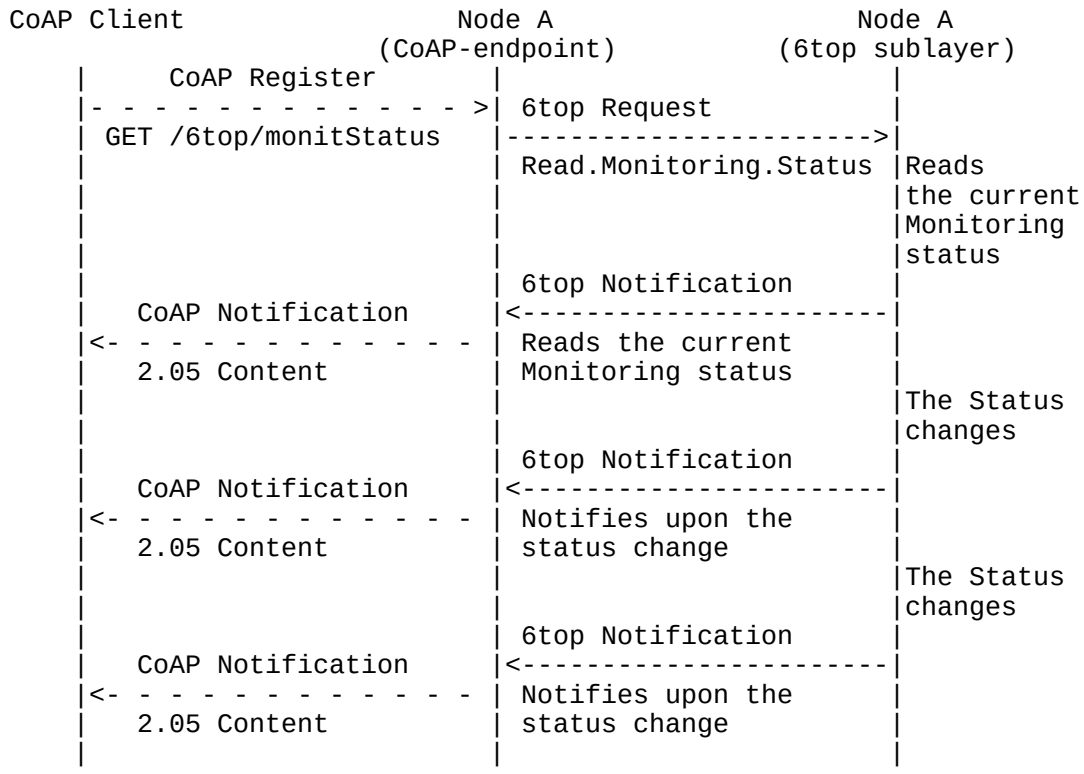


Figure 13: Example of Subscribing to Monitoring Status

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

[I-D.ietf-6tisch-6top-interface]
Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top) Interface", draft-ietf-6tisch-6top-interface-02 (work in progress), October 2014.

[I-D.ietf-6tisch-architecture]
Thubert, P., Watteyne, T., Struik, R., and M. Richardson, "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-architecture-06 (work in progress), March 2015.

[I-D.ietf-6tisch-minimal]
Vilajosana, X. and K. Pister, "Minimal 6TiSCH Configuration", draft-ietf-6tisch-minimal-06 (work in progress), March 2015.

[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-03 (work in progress), January 2015.

[I-D.ietf-6tisch-tsch]
Watteyne, T., Palattella, M., and L. Grieco, "Using IEEE802.15.4e TSCH in an IoT context: Overview, Problem Statement and Goals", draft-ietf-6tisch-tsch-05 (work in progress), January 2015.

[I-D.ietf-core-observe]
Hartke, K., "Observing Resources in CoAP", draft-ietf-core-observe-16 (work in progress), December 2014.

[I-D.vanderstok-core-comi]
Stok, P., Greevenbosch, B., Bierman, A., Schoenwaelder, J., and A. Sehgal, "CoAP Management Interface", draft-vanderstok-core-comi-06 (work in progress), February 2015.

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

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", RFC 7049, October 2013.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, June 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.

Appendix A.

Guidelines for constructing URI path names:

1. The first letter of each element of the path SHOULD be capitalized

2. If an element has multiple words, each the first letter of each work SHOULD be capitalized

Authors' Addresses

Raghuram S Sudhaakar (editor)
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: September 9, 2015

X. Vilajosana, Ed.
Universitat Oberta de Catalunya
K. Pister
University of California Berkeley
March 8, 2015

Minimal 6TiSCH Configuration
draft-ietf-6tisch-minimal-06

Abstract

This document describes the minimal set of rules to operate an IEEE 802.15.4e Timeslotted Channel Hopping (TSCH) network. This minimal mode of operation can be used during network bootstrap, as a fall-back mode of operation when no dynamic scheduling solution is available or functioning, or during early interoperability testing and development.

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 September 9, 2015.

Copyright Notice

Table of Contents

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 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 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 MAY 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, the minimal configuration MAY be used as a fall-back 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 when a node synchronizes to the network. This document describes specific settings for these parameters. Nodes MUST broadcast properly-formed Enhanced Beacons to announce these values.

2. Requirements Language

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

3. 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.

3.1. 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 communicate. This document recommends the following slotframe configuration.

Minimal configuration

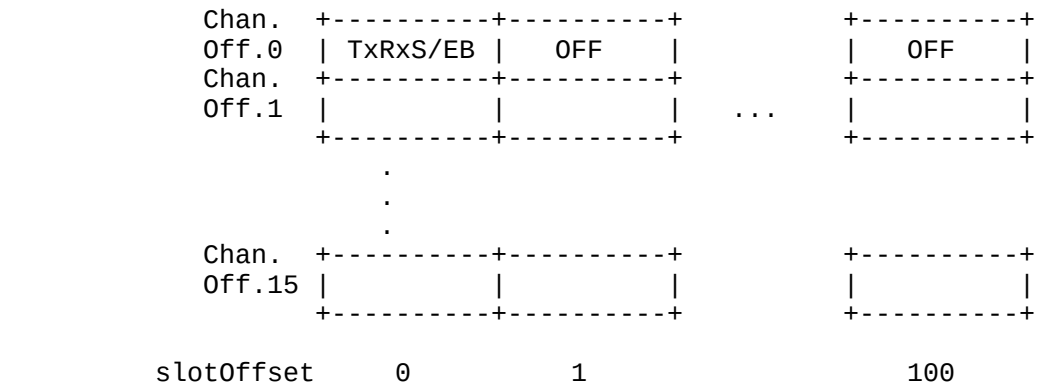
Property	Value
Number of time slots per Slotframe	Variable
Number of available frequencies	16
Number of scheduled cells	1 (slotOffset 0) (macLinkType NORMAL)
Number of unscheduled cells	The remainder of the slotframe
Number of MAC retransmissions (max)	3 (4 attempts to tx)

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/receive both EBs and data link-layer frames. 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 to the link-layer broadcast address (and are therefore not acknowledged). Data packets, as described in Section 3.2, use the same active slot. Per [IEEE802154e], data packets sent unicast on this cell are acknowledged by the receiver. The remaining cells in the slotframe are unscheduled, and MAY be used by dynamic scheduling solutions. Details about such dynamic scheduling solution are out of scope of this document.

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 in the EBs. If one uses a timeslot duration different than 10ms, EBs MUST contain the complete TimeSlot IE as described in Section 3.4. This document

also recommends 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 (MAY be used by a dynamic scheduling mechanism)

3.2. Cell Options

Per [IEEE802154e] TSCH, each scheduled cell has an associated bitmap of cell options, called LinkOptions. The scheduled cell in the minimal schedule is configured as a Hard cell [I-D.ietf-6tisch-tsch][I-D.ietf-6tisch-6top-interface]. Additional available cells MAY 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, listens otherwise. Because the "shared" bit is set, the back-off mechanism defined in [IEEE802154e] is used to resolve contention when transmitting. 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.

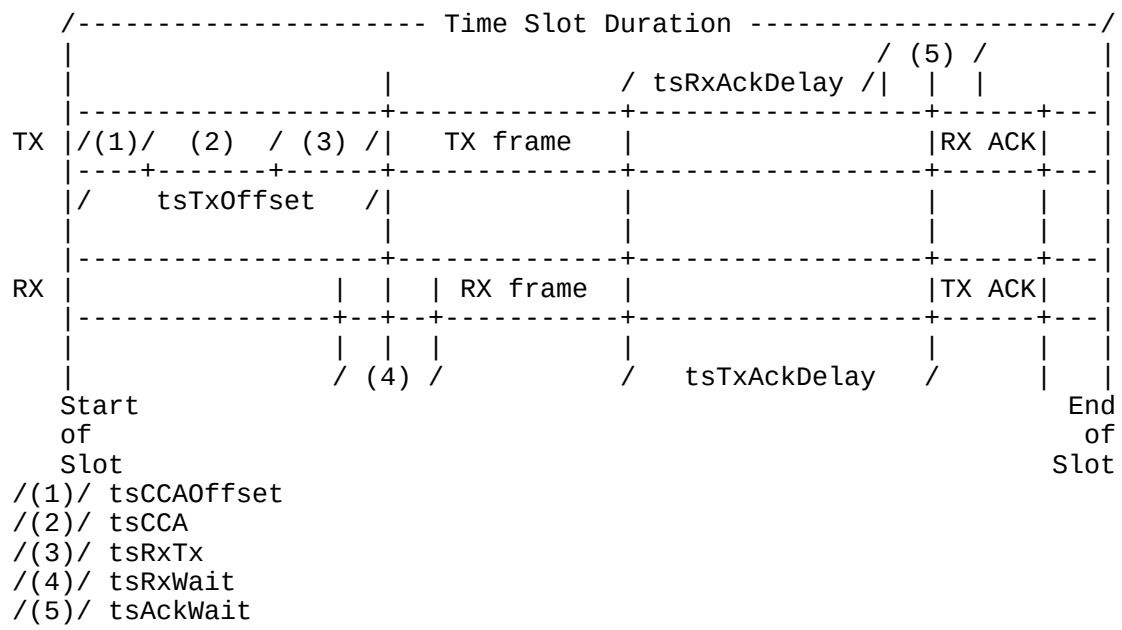
3.3. Retransmissions

The maximum number of link layer retransmissions is set to 3. For packets which require an acknowledgment, 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.

3.4. 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 acknowledgment 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 bit after the Start of Frame Delimiter (SFD) of the transmitted packet leaves the radio of the TX node. The radio of the RX node is turned on tsRxWait/2 before that instant, and listens for at least tsRxWait. This allows for a de-synchronization between the two nodes of at most tsRxWait/2 in either direction (early or late). The RX node needs to send the first bit after the SFD of the MAC acknowledgment exactly TsTxAckDelay after the end of the last byte of the received packet. TX’s radio has to be turned on tsAckWait/2 before that time, and keep listening for at least tsAckWait. The TX node can perform a Clear Channel Assessment (CCA) if required, this does not interfere with the scope of this draft. As for a minimal configuration, CCA is OPTIONAL.

Time slot internal timing diagram



/(1)/ tsCCAOffset
/(2)/ tsCCA
/(3)/ tsRxTx
/(4)/ tsRxWait
/(5)/ tsAckWait

A 10ms time slot length is the default value defined by [IEEE802154e]. Section 6.4.3.3.3 of [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. Note however that in case of discrepancy between the values in this document and [IEEE802154e], the IEEE standard specification has precedence.

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

IEEE802.15.4e TSCH parameter	Value (us)
tsCCAOffset	1800
tsCCA	128
tsTxOffset	2120
tsRxOffset	1120
tsRxAckDelay	800
tsTxAckDelay	1000
tsRxWait	2200
tsAckWait	400
tsRxTx	192
tsMaxAck	2400
tsMaxTx	4256
Time Slot duration	10000

4. Enhanced Beacons Configuration and Content

[IEEE802154e] does not define how often EBs are sent, nor their contents. EBs should not in general be used for synchronization. Synchronization is achieved via acknowledgements to normal packet traffic, and keepalives. 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 are only authenticated and payload is not encrypted. Refer to the 6TiSCH architecture document [I-D.ietf-6tisch-architecture] for further details on security aspects.

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

The content of the IEs is presented here for completeness, however this information is redundant with [IEEE802154e].

4.1. Sync IE

Contains synchronization information such as ASN and Join Priority. The value of Join Priority is discussed in Section 6.2.

4.1.1. IE Header

Length (b0-b7) = 0x06

Sub-ID (b8-b14) = 0x1a

Type (b15) = 0x00 (short)

4.1.2. 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)

4.2. TSCH Timeslot IE

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

4.2.1. IE Header

Length (b0-b7) = 0x01

Sub-ID (b8-b14) = 0x1c

Type (b15) = 0x00 (short)

4.2.2. IE Content

Timeslot Template ID (b0-b7) = 0x00

In the case that a different than the default timeslot template is used, the IE Content MUST follow the following specification as defined in [IEEE802154e], Section 5.2.4.15.

Timeslot Template ID (b0-b7)

macTsCCAOffset (b8-b23)

macTsCCA (b24-b39)

macTsTxOffset (b40-b55)

macTsRxOffset (b56-b71)

macTsRxAckDelay (b72-b87)

macTsTxAckDelay (b88-b103)

macTsRxWait (b104-b119)

macTsAckWait (b120-b135)

macTsRxTx (b136-b151)

macTsMaxAck (b152-b167)

macTsMaxTx (b168-b183)

macTsTimeslotLength (b184-b199)

4.3. 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.

4.3.1. IE Header

Length (b0-b7) = 0x01

Sub-ID (b8-b14) = 0x1d

Type (b15) = 0x00 (short)

4.3.2. IE Content

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

The default sequence for the 2.4GHz OQPSK PHY is [5, 6, 12, 7, 15, 4, 14, 11, 8, 0, 1, 2, 13, 3, 9, 10] per section 5.1.1a of [IEEE802154e]. Note however that in case of discrepancy between the

values in this document and [IEEE802154e], the IEEE standard specification has preference.

4.4. 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 learn the schedule used in the network as they join it.

4.4.1. IE Header

Length (b0-b7) = variable

Sub-ID (b8-b14) = 0x1b

Type (b15) = 0x00 (short)

4.4.2. 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 3.2

5. Acknowledgment

Link-layer acknowledgment frames are built according to [IEEE802154e]. Unicast frames sent to a unicast MAC destination address request an acknowledgment. The sender node MUST set the ACK requested bit in the IEEE802.15.4 header. The acknowledgment frame is of type ACK (0x10). Each acknowledgment contains the following IE:

5.1. ACK/NACK Time Correction IE

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

5.1.1. IE Header

Length (b0-b7) = 0x02

Sub-ID (b8-b14) = 0x1e

Type (b15) = 0x00 (short)

5.1.2. 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

6. Neighbor information

[IEEE802154e] does not define how and when each node in the network keeps information about its neighbors. Keeping the following information in the neighbor table is RECOMMENDED:

6.1. 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. It 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 9.1.1.

6.2. 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 MUST be equivalent to the result of the function DAGRank(rank) as defined by [RFC6550] and Section 9.1.1. 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 higher preference to connect to that device. 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 9.1.1), 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 9.2.3. 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 implementation-specific.

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.

7. Queues and Priorities

[IEEE802154e] does not define the use of queues to handle upper layer data (either application or control data from upper layers). The use of a single queue with the following rules is RECOMMENDED:

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 queuing priority than packets received from a higher layer.

IEEE802.15.4 frame types Beacon and Command have a higher queuing priority than IEEE802.15.4 frame 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.

8. Security

As this document refers to the interaction between Layer 3 and Layer 2 protocols, this interaction MUST be secured by L2 security mechanisms as defined by [IEEE802154e]. Two security mechanisms are considered, authentication and encryption, authentication applies to the all packet content while encryption applies to header IEs and MAC payload. Key distribution is out of scope of this document, but examples include pre-configured keys at the nodes, shared keys among peers or well-known keys. Refer to the 6TiSCH architecture document [I-D.ietf-6tisch-architecture] for further details on key distribution and advanced security aspects.

The present document assumes the existence of two keys, which can be well-known by the network devices and/or pre-configured. One of the keys (K1) is used to authenticate EBs (all frame). As defined in Section 4 EBs MUST be authenticated but payload not encrypted. This prevents two independent networks to interfere or enable non-allowed nodes to join a particular network. A second key (K2) is used to authenticate and encrypt the payload of DATA, ACKNOWLEDGEMENT, MAC COMMAND frame types and respective header IEs.

9. RPL on TSCH

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

9.1. RPL Objective Function Zero

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

9.1.1. Rank computation

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

R(N) = R(P) + rank_increment

rank_increment = (Rf*Sp + Sr) * MinHopRankIncrease

Where:

R(N): Rank of the node.

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

rank_increment: The result of a function that determines the rank increment.

Rf (rank_factor): A configurable factor that is used to multiply the effect of the link properties in the rank_increment computation. If none is configured, rank_factor of 1 is used. In this specification, a 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

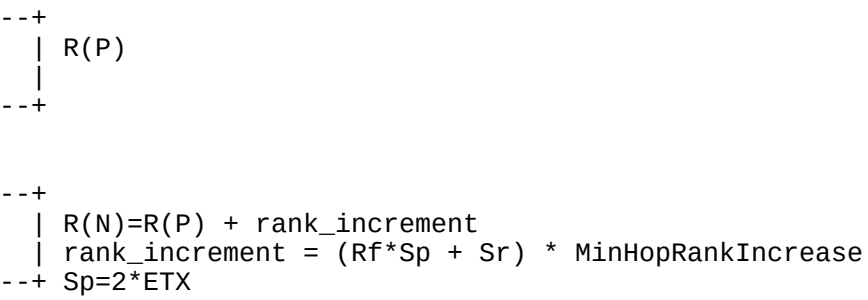
the number of transmitted packets to a certain node. E.g: Sp=2*numTX/numTXAck

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_increment [RFC6550]. DEFAULT_MIN_HOP_rank_increment 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 floor(x) where floor(x) is the function that evaluates to the greatest integer less than or equal to x. DAGRank(rank) = floor(rank/MinHopRankIncrease)

Rank computation scenario



9.1.2. Rank computation Example

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

Rf = 1

Sp = 2* ETX

Sr = 0

minHopRankIncrease = 256 (default in RPL)

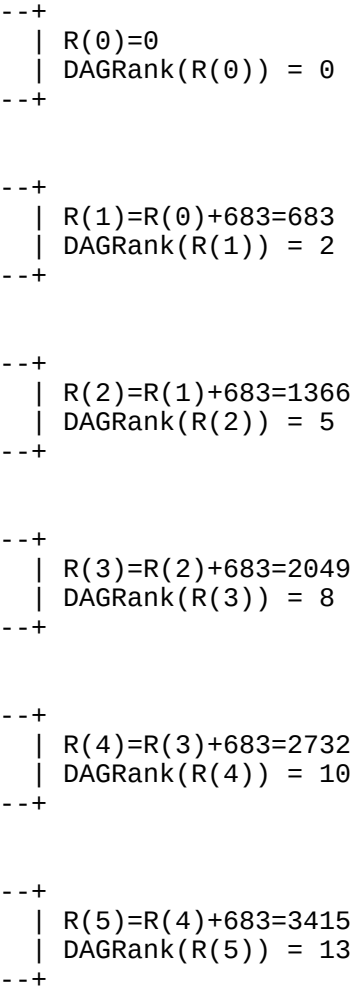
```
ETX=(numTX/numTXAck)

r(n) = r(p) + rank_increment

rank_increment = (Rf*Sp + Sr) * minHopRankIncrease

rank_increment = 512*numTx/numTxACK
```

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



9.2. RPL Configuration

In addition to the Objective Function (OF), a minimal configuration for RPL SHOULD indicate the preferred mode of operation (either Storing Mode or Non-Storing Mode) so different RPL implementations can inter-operate. RPL information and hop-by-hop extension headers MUST follow [RFC6553] and [RFC6554] specification. In the case that the packets formed at the LLN need to cross through intermediate routers, these MUST obey to the IP in IP encapsulation requirement specified by the [RFC6282] and [RFC2460]. RPI and RH3 extension headers and inner IP headers MUST be compressed according to [RFC6282].

9.2.1. 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.

9.2.2. 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].

9.2.3. 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*ETX. This is mechanism is suited to deal with parent hysteresis in both cases routing parent and time source neighbor selection.

9.2.4. 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

10. Acknowledgments

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

11. References

11.1. Normative References

[RFC6719] Gnawali, O. and P. Levis, "The Minimum Rank with Hysteresis Objective Function", RFC 6719, September 2012.

[RFC6282] Hui, J. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, September 2011.

[RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, 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.

[RFC6552] Thubert, P., "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6552, 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.

[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.

[RFC6206] Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko, "The Trickle Algorithm", RFC 6206, March 2011.

[RFC3610] Whiting, D., Housley, R., and N. Ferguson, "Counter with CBC-MAC (CCM)", RFC 3610, September 2003.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

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

11.2. Informative References

[I-D.ietf-6tisch-tsch]
Watteyne, T., Palattella, M., and L. Grieco, "Using IEEE802.15.4e TSCH in an IoT context: Overview, Problem Statement and Goals", draft-ietf-6tisch-tsch-05 (work in progress), January 2015.

[I-D.ietf-6tisch-architecture]
Thubert, P., Watteyne, T., Struik, R., and M. Richardson, "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-architecture-05 (work in progress), January 2015.

[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-03 (work in progress), January 2015.

[I-D.ietf-6tisch-6top-interface]
Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top) Interface", draft-ietf-6tisch-6top-interface-02 (work in progress), October 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.

11.3. 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.

[CCM] National Institute of Standards and Technology, "Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality. SP 800-38C", May 2004.

[CCM-Star]
Struik, R., "Formal Specification of the CCM* Mode of Operation, IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs).", September 2005.

[decouti03high]
De Couto, D., Aguayo, D., Bicket, J., and R. Morris, "A High-Throughput Path Metric for Multi-Hop Wireless Routing", ACM International Conference on Mobile Computing and Networking (MobiCom) , 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: September 3, 2018

MR. Palattella, Ed.
LIST
P. Thubert
cisco
T. Watteyne
Analog Devices
Q. Wang
Univ. of Sci. and Tech. Beijing
March 2, 2018

Terms Used in IPv6 over the TSCH mode of IEEE 802.15.4e
draft-ietf-6tisch-terminology-10

Abstract

This document provides a glossary of terminology used in IPv6 over the TSCH mode of IEEE 802.15.4e (6TiSCH). This document extends existing terminology documents for Low-power and Lossy Networks.

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 https://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 September 3, 2018.

Copyright Notice

Table of Contents

1. Introduction

The IEEE802.15.4 Medium Access Control (MAC) has evolved with the Time Slotted Channel Hopping (TSCH) mode for industrial-type applications.

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 [RFC7102] and use terms from [RFC6550] and [RFC6552], which are all included here by reference.

The draft does not reuse terms from IEEE802.15.4 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 TSCH mode of IEEE 802.15.4e): It defines the 6top sublayer, a set of protocols for setting up a TSCH schedule in distributed approach, and a security solution.

6top (6TiSCH Operation Sublayer): The next highest layer of the IEEE802.15.4 TSCH medium access control layer. It implements and terminates 6P, and contains at least one SF.

6P (6top Protocol): Allows neighbor nodes to communicate to add/ delete cells to one another in their TSCH schedule.

6P Transaction: Part of 6P, the action of two neighbors exchanging a 6P request message and the corresponding 6P response message.

ASN (Absolute Slot Number): The total number of timeslots that have elapsed since the PAN coordinator has started the TSCH network. Incremented by one at each timeslot. It is wide enough to not roll over in practice.

BBR (Backbone Router): An LBR and also a IPv6 ND-efficiency-aware Router (NEAR) [I-D.chakrabarti-nordmark-6man-efficient-nd]. Performs ND proxy operations between registered devices and classical ND devices that are located on the backbone.

blacklist of frequencies: A set of frequencies which should not be used for communication.

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. For a given slotframe length, the size of the bundle translates directly into bandwidth. A bundle is a local abstraction that represents a half-duplex link for either sending or receiving, with bandwidth that amounts to the sum of the cells in the bundle.

CCA (Clear Channel Assessment): Mechanism defined in [IEEE802154-2015], section 6.2.5.2. In a TSCH network, CCA can be used to detect other radio networks in vicinity. Nodes listen the channel before sending, to detect other ongoing transmissions. Because the network is synchronized, CCA cannot be used to detect colliding transmission within the same network.

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

centralized cell reservation: A reservation of a cell done by a centralized entity (e.g., a PCE) in the network.

centralized track reservation: A reservation of a track done by a centralized entity (e.g., a PCE) in the network.

Channel Distribution/Usage (CDU) matrix: : Matrix of cells (i,j) representing the spectrum (channel) distribution among the different nodes in the 6TiSCH network. The CDU matrix has width in timeslots, equal to the period of the network scheduling operation, and height equal to the number of available channels. Every cell (i,j) in the CDU, identified by (slotOffset, channelOffset), belongs to a specific chunk. It has to be noticed that such a matrix which includes all the cells grouped in chunks, belonging to different slotframes, is different from the TSCH schedule.

channelOffset: Identifies a row in the TSCH schedule. The number of available channelOffset values is equal to the number of available frequencies. The channelOffset translates into a frequency when the communication takes place, resulting in channel hopping.

chunk: A well-known list of cells, distributed in time and frequency, within a CDU matrix. A chunk represents a portion of a CDU matrix. The partition of the CDU matrix in chunks is globally known by all the nodes in the network to support the appropriation process, which is a negotiation between nodes within an interference domain. A node that manages to appropriate a chunk gets to decide which transmissions will occur over the cells in the chunk within its interference domain (i.e., a parent node will decide when the cells within the appropriated chunk are used and by which node, among its children.

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

deterministic network: The generic concept of deterministic network is defined in [I-D.ietf-detnet-architecture]. When applied to 6TiSCH, it refers to the reservation of tracks which guarantee an end-to-end latency and optimize the PDR for well-characterized flows.

distributed cell reservation: A reservation of a cell done by one or more in-network entities.

distributed track reservation: A reservation of a track done by one or more in-network entities.

EB (Enhanced Beacon): A special frame defined used by a node, including the JP, to announce the presence of the

network. It contains enough information for a pledge to synchronize to the network.

hard cell: A scheduled cell which the 6top sublayer cannot relocate.

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.

IE (Information Element): Type-Length-Value containers placed at the end of the MAC header, used to pass data between layers or devices. Some IE identifiers are managed by the IEEE [IEEE802154-2015]. Some IE identifiers are managed by the IETF [I-D.kivinen-802-15-ie].

join process: The overall process that includes the discovery of the network by pledge(s) and the execution of the join protocol.

join protocol: The protocol that allows the pledge to join the network. The join protocol encompasses authentication, authorization and parameter distribution. The join protocol is executed between the pledge and the JRC.

joined node: The new device, after having completed the join process, often just called a node.

JP (Join Proxy): Node already part of the 6TiSCH network that serves as a relay to provide connectivity between the pledge and the JRC. The JP announces the presence of the network by regularly sending EB frames.

JRC (Join Registrar/Coordinator): Central entity responsible for the authentication, authorization and configuration of the pledge.

LBR: Low-power Lossy Network (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, the layer immediately below IP. The IETF parlance for the term "Link" is adopted, as opposed to the IEEE802.15.4 terminology.

pledge: A new device that attempts to join a 6TiSCH network.

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

(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 IEEE802.15.4 TSCH implementation to communicate. A scheduled cell can either be a hard or a soft cell.

SF (6top Scheduling Function): The cell management entity that adds or deletes cells dynamically based on application networking requirements. The cell negotiation with a neighbor is done using 6P.

SFID (6top Scheduling Function Identifier): A 4-bit field identifying an SF.

shared cell: A cell marked with both the "TX" and "shared" flags. This cell can be used by more than one transmitter node. A back-off algorithm is used to resolve contention.

slotframe: A collection of timeslots repeating in time, analogous to a superframe in that it defines periods of communication opportunities. 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.

slotOffset: A column in the TSCH schedule, i.e. the number of timeslots since the beginning of the current iteration of the slotframe.

soft cell: A scheduled cell which the 6top sublayer can relocate.

time source neighbor: A neighbor that a node uses as its time reference, and to which it needs to keep its clock synchronized.

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.

track: A determined sequence of cells along a multi-hop path. It is typically the result of a track reservation. The node that initializes the process of 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.

TSCH (6top Scheduling Function Identifier): A medium access mode of the [IEEE802154-2015] 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 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 IEEE802.15.4 TSCH implementation.

3. Security Considerations

Since this document specifies terminology and does not specify new procedures or protocols, it raises no new security issues.

4. References

4.1. Normative References

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

[RFC6552] Thubert, P., Ed., "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6552, DOI 10.17487/RFC6552, March 2012, <https://www.rfc-editor.org/info/rfc6552>.

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

4.2. 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-07 (work in progress), February 2015.

[I-D.ietf-detnet-architecture]
Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-04 (work in progress), October 2017.

[I-D.kivinen-802-15-ie]
Kivinen, T. and P. Kinney, "IEEE 802.15.4 Information Element for IETF", draft-kivinen-802-15-ie-06 (work in progress), March 2017.

4.3. External Informative References

[IEEE802154-2015]
IEEE standard for Information Technology, "IEEE Std 802.15.4-2015 Standard for Low-Rate Wireless Personal Area Networks (WPANs)", December 2015.

Authors' Addresses

Maria Rita Palattella (editor)
Luxembourg Institute of Science and Technology
Department 'Environmental Research and Innovation' (ERIN)
41, rue du Brill
Belvaux L-4422
Luxembourg

Phone: (+352) 275 888-5055
Email: mariarita.palattella@list.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
Analog Devices
32990 Alvarado-Niles Road, Suite 910
Union City, CA 94587
USA

Email: thomas.watteyne@analog.com

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

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

6TiSCH
Internet-Draft
Intended status: Informational
Expires: September 10, 2015

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

March 9, 2015

Using IEEE802.15.4e TSCH in an IoT context: Overview, Problem Statement and Goals draft-ietf-6tisch-tsch-06

Abstract

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

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

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

This Internet-Draft will expire on September 10, 2015.

Copyright Notice

Table of Contents

1. Introduction

IEEE802.15.4e [IEEE802154e] was published in 2012 as an amendment to the Medium Access Control (MAC) protocol defined by the IEEE802.15.4-2011 [IEEE802154] standard. IEEE802.15.4e will be rolled into the next revision of IEEE802.15.4, scheduled to be published in 2015. The Timeslotted Channel Hopping (TSCH) mode of IEEE802.15.4e is the object of this document.

This document describes the main issues arising from the adoption of the IEEE802.15.4e TSCH in the LLN context, following the terminology defined in [I-D.ietf-6tisch-terminology]. Appendix A further gives an overview of the key features of the IEEE802.15.4e Timeslotted Channel Hopping (TSCH) amendment. Appendix B details features of IEEE802.15.4e TSCH which might be interesting for the work of the 6TiSCH WG.

TSCH was designed to allow IEEE802.15.4 devices to support a wide range of applications including, but not limited to, industrial ones [IEEE802154e]. At its core is a medium access technique which uses time synchronization to achieve low power operation and channel hopping to enable high reliability. Synchronization accuracy impacts power consumption, and can vary from micro-seconds to milli-seconds depending on the solution. This is very different from the "legacy" IEEE802.15.4 MAC protocol, and is therefore better described as a "redesign". TSCH does not amend the physical layer; i.e., it can operate on any IEEE802.15.4-compliant hardware.

IEEE802.15.4e is the latest generation of ultra-lower power and reliable networking solutions for LLNs. [RFC5673] discusses industrial applications, and highlights the harsh operating conditions as well as the stringent reliability, availability, and security requirements for an LLN to operate in an industrial environment. In these environments, vast deployment environments with large (metallic) equipment cause multi-path fading and interference to thwart any attempt of a single-channel solution to be reliable; the channel agility of TSCH is the key to its ultra high reliability. Commercial networking solutions are available today in which nodes consume 10’s of micro-amps on average [CurrentCalculator] with end-to-end packet delivery ratios over 99.999% [doherty07channel].

IEEE802.15.4e has been designed for low-power constrained devices, often called "motes". Several terms are used in the IETF to refer to those devices, including "LLN nodes" [RFC7102] and "constrained nodes" [RFC7228]. In this document, we use the generic (and shorter) term "node", used as a synonym for "LLN node", "constrained node" or "mote".

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

IEEE802.15.4e TSCH focuses on the MAC layer only. This clean layering allows for TSCH to fit under an IPv6 enabled protocol stack for LLNs, running 6LoWPAN [RFC6282], IPv6 Routing Protocol for Low power and Lossy Networks (RPL) [RFC6550] and the Constrained Application Protocol (CoAP) [RFC7252]. What is missing is a functional entity which is in charge of scheduling TSCH timeslots for frames to be sent on. In this document, we refer to this entity as the "Logical Link Control" (LLC), bearing in mind that realizations of this entity can be of different types, including a distributed protocol or a centralized server in charge of scheduling.

While [IEEE802154e] defines the mechanisms for a TSCH node to communicate, it does not define the policies to build and maintain the communication schedule, match that schedule to the multi-hop paths maintained by RPL, adapt the resources allocated between neighbor nodes to the data traffic flows, enforce a differentiated treatment for data generated at the application layer and signaling messages needed by 6LoWPAN and RPL to discover neighbors, react to topology changes, self-configure IP addresses, or manage keying material.

In other words, IEEE802.15.4e TSCH is designed to allow optimizations and strong customizations, simplifying the merging of TSCH with a protocol stack based on IPv6, 6LoWPAN, and RPL.

2. TSCH in the LLN Context

To map the services required by the IP layer to the services provided by the link layer, an adaptation layer is used [palattella12standardized]. The 6LoWPAN working group started working in 2007 on specifications for transmitting IPv6 packets over IEEE802.15.4 networks [RFC4919]. Low-power Wireless Personal Area Networks (WPANs) typically feature small frame sizes, support for addresses with different lengths, low bandwidth, star and mesh topologies, battery powered devices, low cost, large number of devices, unknown node positions, high unreliability, and periods during which communication interfaces are turned off to save energy. Given these features, it is clear that the adoption of IPv6 on top of a Low-Power WPAN is not straightforward, but poses strong requirements for the optimization of this adaptation layer.

For instance, due to the IPv6 default minimum MTU size (1280 bytes), an un-fragmented IPv6 packet is too large to fit in an IEEE802.15.4 frame. Moreover, the overhead due to the 40-byte long IPv6 header wastes the scarce bandwidth available at the PHY layer [RFC4944]. For these reasons, the 6LoWPAN working group has defined an effective adaptation layer [RFC6282]. Further issues encompass the auto-configuration of IPv6 addresses [RFC2460][RFC4862], the compliance with the recommendation on supporting link-layer subnet broadcast in shared networks [RFC3819], the reduction of routing and management overhead [RFC6606], the adoption of lightweight application protocols (or novel data encoding techniques), and the support for security mechanisms (confidentiality and integrity protection, device bootstrapping, key establishment, and management).

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

Routing issues are challenging for 6LoWPAN, given the low-power and lossy radio links, the battery-powered nodes, the multi-hop mesh topologies, and the frequent topology changes due to mobility. Successful solutions take into account the specific application requirements, along with IPv6 behavior and 6LoWPAN mechanisms [palattella12standardized]. The ROLL working group has defined RPL in [RFC6550]. RPL can support a wide variety of link layers, including ones that are constrained, potentially lossy, or typically utilized in conjunction with host or router devices with very limited resources, as in building/home automation [RFC5867][RFC5826], industrial environments [RFC5673], and urban applications [RFC5548]. RPL is able to quickly build up network routes, distribute routing knowledge among nodes, and adapt to a changing topology. In a typical setting, nodes are connected through multi-hop paths to a small set of root devices, which are usually responsible for data collection and coordination. For each of them, a Destination Oriented Directed Acyclic Graph (DODAG) is created by accounting for link costs, node attributes/status information, and an Objective Function, which maps the optimization requirements of the target scenario.

The topology is set up based on a Rank metric, which encodes the distance of each node with respect to its reference root, as specified by the Objective Function. Regardless of the way it is computed, the Rank monotonically decreases along the DODAG towards the root, building a gradient. RPL encompasses different kinds of traffic and signaling information. Multipoint-to-Point (MP2P) is the dominant traffic in LLN applications. Data is routed towards nodes with some application relevance, such as the LLN gateway to the larger Internet, or to the core of private IP networks. In general, these destinations are the DODAG roots and act as data collection

points for distributed monitoring applications. Point-to-Multipoint (P2MP) data streams are used for actuation purposes, where messages are sent from DODAG roots to destination nodes. Point-to-Point (P2P) traffic allows communication between two devices belonging to the same LLN, such as a sensor and an actuator. A packet flows from the source to the common ancestor of those two communicating devices, then downward towards the destination. RPL therefore has to discover both upward routes (i.e. from nodes to DODAG roots) in order to enable MP2P and P2P flows, and downward routes (i.e. from DODAG roots to nodes) to support P2MP and P2P traffic.

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

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

3. Problems and Goals

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

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

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

3.1. Network Formation

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

1. Define the Information Elements included in the Enhanced Beacons [IEEE802154e] advertising the presence of the network.

2. For a new node, define rules to process and filter received Enhanced Beacons.

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

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

3.2. Network Maintenance

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

1. Manage each node's time source neighbor.

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

3. Schedule transmissions of Enhanced Beacons to advertise the presence of the network.

3.3. Multi-Hop Topology

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

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

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

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

3.4. Routing and Timing Parents

At all times, a TSCH node needs to have a time source neighbor it can synchronize to. The LLC therefore needs to assign a time source neighbor to allow for correct operation of the TSCH network. A time

source neighbors could, or not, be taken from the RPL routing parent set.

3.5. Resource Management

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

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

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

3.6. Dataflow Control

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

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

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

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

3.7. Deterministic Behavior

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

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

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

3.8. Scheduling Mechanisms

Several scheduling mechanisms can be envisioned, and possibly coexist in the same network. For example, [I-D.phinney-roll-rpl-industrial-applicability] describes how the allocation of bandwidth can be optimized by an external Path Computation Element (PCE) [RFC4655]. Another centralized (PCE-based) traffic-aware scheduling algorithm is defined in [TASA-PIMRC]. Alternatively, two neighbor nodes can adapt the number of cells autonomously by monitoring the amount of traffic, and negotiating the allocation to extra cell when needed. An example of decentralized algorithm (i.e. no PCE is needed) is provided in [tinka10decentralized]. This mechanism can be used to establish multi-hop paths in a fashion similar to RSVP [RFC2205]. The LLC needs to:

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

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

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

3.9. Secure Communication

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

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

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

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

4. IANA Considerations

This memo includes no request to IANA.

5. Security Considerations

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

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

6. Acknowledgments

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

7. References

7.1. Normative 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.

7.2. Informative References

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, June 2014.

[RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, May 2014.

[RFC7102] Vasseur, JP., "Terms Used in Routing for Low-Power and Lossy Networks", RFC 7102, January 2014.

[RFC6606] Kim, E., Kaspar, D., Gomez, C., and C. Bormann, "Problem Statement and Requirements for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing", RFC 6606, May 2012.

[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.

[RFC6282] Hui, J. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, September 2011.

[RFC5867] Martocci, J., De Mil, P., Riou, N., and W. Vermeylen, "Building Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5867, June 2010.

[RFC5826] Brandt, A., Buron, J., and G. Porcu, "Home Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5826, April 2010.

[RFC5673] Pister, K., Thubert, P., Dwars, S., and T. Phinney, "Industrial Routing Requirements in Low-Power and Lossy Networks", RFC 5673, October 2009.

[RFC5548] Dohler, M., Watteyne, T., Winter, T., and D. Barthel, "Routing Requirements for Urban Low-Power and Lossy Networks", RFC 5548, May 2009.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, September 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.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.

[RFC3819] Karn, P., Bormann, C., Fairhurst, G., Grossman, D., Ludwig, R., Mahdavi, J., Montenegro, G., Touch, J., and L. Wood, "Advice for Internet Subnetwork Designers", BCP 89, RFC 3819, July 2004.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC2205] Braden, B., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.

[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-03 (work in progress), January 2015.

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

[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.

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

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

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

[CurrentCalculator] Linear Technology, "Application Note: Using the Current Calculator to Estimate Mote Power", August 2012, <http://www.linear.com/docs/42452>.

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

[tinka10decentralized]
Tinka, A., Watteyne, T., and K. Pister, "A Decentralized Scheduling Algorithm for Time Synchronized Channel Hopping", Ad Hoc Networks 2010, 2010.

[watteyne09reliability]
Watteyne, T., Mehta, A., and K. Pister, "Reliability Through Frequency Diversity: Why Channel Hopping Makes Sense", International Conference on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN) 2009, Oct. 2009.

[TASA-PIMRC]
Palattella, MR., Accettura, N., Dohler, M., Grieco, LA., and G. Boggia, "Traffic Aware Scheduling Algorithm for Multi-Hop IEEE 802.15.4e Networks", IEEE PIMRC 2012, Sept. 2012.

[palattella12standardized]
Palattella, MR., Accettura, N., Vilajosana, X., Watteyne, T., Grieco, LA., Boggia, G., and M. Dohler, "Standardized Protocol Stack For The Internet Of (Important) Things", IEEE Communications Surveys and Tutorials 2012, Dec. 2012.

Appendix A. TSCH Protocol Highlights

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

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

A.1. Timeslots

All nodes in a TSCH network are synchronized. Time is sliced up into timeslots. A timeslot is long enough for a MAC frame of maximum size to be sent from node A to node B, and for node B to reply with an acknowledgment (ACK) frame indicating successful reception.

The duration of a timeslot is not defined by the standard. With IEEE802.15.4-compliant radios operating in the 2.4GHz frequency band, a maximum-length frame of 127 bytes takes about 4ms to transmit; a

shorter ACK takes about 1ms. With a 10ms slot (a typical duration), this leaves 5ms to radio turnaround, packet processing and security operations.

A.2. Slotframes

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

A.3. Node TSCH Schedule

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

Once a node obtains its schedule, it executes it:

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

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

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

A.4. Cells and Bundles

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

A single element of the schedule characterized by a slotOffset and channelOffset, and reserved for node A to transmit to node B (or for

node B to receive from node A) within a given slotframe, is called a "scheduled cell".

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

A.5. Dedicated vs. Shared Cells

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

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

A.6. Absolute Slot Number

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

ASN = (k*S+t)

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

A.7. Channel Hopping

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

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

The function F consists of a look-up table containing the set of available channels. The value nFreq (the number of available frequencies) is the size of this look-up table. There are as many channelOffset values as there are frequencies available (e.g. 16 when using IEEE802.15.4-compliant radios at 2.4GHz, when all channels are used). Since both nodes have the same channelOffset written in their schedule for that scheduled cell, and the same ASN counter, they compute the same frequency. At the next iteration (cycle) of the slotframe, however, while the channelOffset is the same, the ASN has changed, resulting in the computation of a different frequency.

This results in "channel hopping": even with a static schedule, pairs of neighbors "hop" between the different frequencies when communicating. A way of ensuring communication happens on all available frequencies is to set the number of timeslots in a slotframe to a prime number. Channel hopping is a technique known to efficiently combat multi-path fading and external interference [watteyne09reliability].

A.8. Time Synchronization

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

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

TSCH adds timing information in all packets that are exchanged (both data and ACK frames). This means that neighbor nodes can

resynchronize to one another whenever they exchange data. In detail, two methods are defined in IEEE802.15.4e-2012 for allowing a device to synchronize in a TSCH network: (i) Acknowledgment-Based and (ii) Frame-Based synchronization. In both cases, the receiver calculates the difference in time between the expected time of frame arrival and its actual arrival. In Acknowledgment-Based synchronization, the receiver provides such information to the sender node in its acknowledgment. In this case, it is the sender node that synchronizes to the clock of the receiver. In Frame-Based synchronization, the receiver uses the computed delta for adjusting its own clock. In this case, it is the receiver node that synchronizes to the clock of the sender.

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

A.9. Power Consumption

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

A.10. Network TSCH Schedule

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

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

A.11. Join Process

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

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

The IEEE802.15.4e TSCH standard does not define the steps beyond this network "bootstrap".

A.12. Information Elements

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

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

A.13. Extensibility

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

An intermediate layer can be defined to manage the MAC layer by defining new Payload IEs.

Appendix B. TSCH Features

This section details features of IEEE802.15.4e TSCH which might be interesting for the work of the 6TiSCH WG. It does not define any requirements.

B.1. Collision Free Communication

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

B.2. Multi-Channel vs. Channel Hopping

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

B.3. Cost of (continuous) Synchronization

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

B.4. Topology Stability

The channel hopping nature of TSCH causes links to be very "stable". Wireless phenomena such as multi-path fading and external interference impact a wireless link between two nodes differently on each frequency. If a transmission from node A to node B fails, retransmitting on a different frequency has a higher likelihood of succeeding that retransmitting on the same frequency. As a result, even when some frequencies are "behaving bad", channel hopping "smoothens" the contribution of each frequency, resulting in more stable links, and therefore a more stable topology.

B.5. Multiple Concurrent Slotframes

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

1. Transmissions take precedence over receptions;

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

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

Authors' Addresses

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

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

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

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

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

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

6tisch
Internet-Draft
Intended status: Informational
Expires: July 13, 2015

R. Struik
Struik Security Consultancy

January 9, 2015

6TiSCH Security Architectural Considerations
draft-struik-6tisch-security-considerations-01

Abstract

This document describes 6TiSCH security architectural elements with high level requirements and the security framework that are relevant for the design of the 6TiSCH security solution.

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 July 13, 2015.

Copyright Notice

Table of Contents

1. Join Protocol Behavior

1.1. MAC Behavior

1. The joining node has to transgress from the so-called "embryonic stage", where it does not have shared keying material with any network nodes yet, to the stage where it has shared keying material with the security manager of the network (who hands out a network wide key, amongst other things). In many cases, the security manager will be the PAN coordinator.

2. Initially, the joining node listens to an enhanced beacon sent by its neighbor node. If this beacon is secured, it can still extract the visible portion of the enhanced beacon frame (which includes all frame fields before these were secured by the neighbor node if the frame was authenticated and which includes only the header fields, including potential header information elements, otherwise). With 802.15.4-2011, the passive scan procedure supports this (see 5.1.2.1.2). In either case, the joining node stores the PAN Descriptor. Note that it cannot rely on the authenticity of the PAN Descriptor, since the beacon frame is either not secured, or it was secured and the joining node did not have a shared key. Either way, it has to accept the PAN Descriptor "on face value".

3. The neighbor node, if it operates securely, normally does not accept incoming frames from the joining node, since these would not be properly secured with the correct keying material. However, the 802.15.4 specification allows one exception to this: it also accepts incoming messages from specifically identified devices that have diplomatic immunity (have so-called "exempt status"). This mechanism can be used to facilitate communication between a joining node and a neighbor node till they have

established shared keying material (whereby the joining node can emerge out of its initial embryonic stage). This can be done as follows:

* The neighbor node can temporarily give the joining node "exempt status", e.g., after failed incoming security processing (thereby, allowing subsequent unsecured data frames from this joining node to be accepted *from this specific device*). It can also populate the table with exempt devices via other means.

* The higher layer can switch on/off this "exempt status" facility for specific joining nodes based on local criteria (one joining node at the time; device open for enrollment of devices or not, pre-populated table, etc.)

* The higher layer of the neighbor node should ensure that this facility is only used for MAC data frames that correspond to initial join messages.

* The higher layer can use this "exempt status" flag for outgoing messages back to the joining node (where this indicates "please send message unsecured" (since message to newbee joining node with diplomatic immunity status).

4. Once the joining node and the neighbor node have established a shared key, the neighbor node can lift the diplomatic immunity status of the joining node (by removing the "exempt status" flag corresponding to this device), after which it may only accept incoming messages from the joining node if these are properly secured. Conversely, the joining node can now update its security policy settings, after which it may only accept properly secured messages received from the neighbor node. Note that from that moment on, the communications between the joining node and the neighbor node can all be authenticated, including time corrections that are very important for proper operation of TSCH (where, e.g., neighbor node is time "clock tower" for joining node).

5. Conceptually, the use of the "exempt flag" could be considered as a mechanism for forming a temporary two-node "join network" (consisting of the joining node and its neighbor node), in which join-related messages are allowed to flow unsecurely. This does not mean, however, that these nodes operate in a separate PAN, though, since incoming frame processing relies on filtering on a single destination PAN Identifier (see 802.15.4-2011, 5.1.6.2), which implies that the neighbor node can only be part of a single PAN (802.15.4-2011 does not know the concept of "multiple PAN

instances"). This also implies that there is no mechanism within 802.15.4 to designate frames for "join" purposes or other special uses (as Wireless HART seems to do with enhanced beacon frames). Of course, there are ways to still artificially realize this, e.g., based on context information (overloading semantics of schedules) or based on yet-to-be-defined information elements (so as to make these act as frame "sub-types"), should one wish to emulate this behavior. Emulating any of this would require changes to 802.15.4 security processing. Currently, there does not seem to be a need for this additional complexity.)

1.2. MAC Security Considerations

1. With 802.15.4-2011, incoming security processing requires access to device-specific information of the originating device (stored on the recipient device in the so-called device descriptor table). This includes the extended address of the originating device, the "lowest" unseen frame counter for that device, and its "exempt status". Successful incoming security processing of a secured frame results in a state change of this device-specific information (since this updates, e.g., the frame counter).

2. Successful incoming security processing of a secured or unsecured frame may result in other state changes as well if only because the device simply "acts" on the received frame or, e.g., due to side effects of the successful receipt hereof. Examples of such side effects include actions triggered by information elements contained in the received frame, such as time corrections to the local clock (which are very important for proper operation of TSCH).

3. 802.15.4-2011 uses the AEAD scheme CCM for frame security, where the nonce is derived from the frame counter and other information. The security of this scheme (or other nonce-based authenticated encryption scheme) is void if nonces are ever reused with the same key. We give an example illustrating how nonce reuse breaks confidentiality: one can derive from two ciphertexts the xor of the corresponding plaintext (or the segment with the size of the shorter ciphertext). From this information and side information on the plaintext (e.g., redundancy), one can often recover both plaintexts (with virtually no remaining ambiguity).

4. Since successful incoming security processing induces a state change, it is imperative that all cryptographic keys used are, indeed, real keys. In particular, this implies that one shall never use 802.15.4 with "default" keys (fake keys with an easy to guess, low-entropy value).

5. If a device wants to communicate with a corresponding party with which it does not share cryptographic keying material yet (e.g., because it is a joining node in embryonic stage), it should send unsecured frames and *not* frames *obscured* (via security through obscurity techniques) using "fake" keys, if only because of avoidance of undesirable side effects: if a recipient accepts an unsecured frame (e.g, because the originator has "exempt status"), this does *not* trigger a state change of security-relevant parameters, whereas if a recipient accepts an obscured frame (secured using a "fake" key), this *does* trigger a state change of security-relevant parameters.

6. TSCH security with 802.15.4e-2012 relies on nonces that are derived from the absolute slot number (ASN), rather than from the frame counter in the device descriptor. Successul processing of a secured incoming frame depends on both originator and recipient of the frame having synchronized "world views" of the ASN entry. The ASN is also used for communication purposes, since indicates scheduling information. This "mixed" use (both for communication and security) is somewhat problematic, since changes to this parameter for either use has spill-over effects on the other use: any changes to the ASN as a communication parameter now might have side effects on security-critical parameters that could, worst case, entirely break security; conversely, any changes to the ASN as a security parameter, e.g., resulting from its inadverent use with a compromised key (or, equivalently, a "fake" key), could result in unreliability of this parameter for indicating scheduling information. Impact of ASN manipulation on security may include reuse of nonces (resulting in compromise of the AEAD cipher’s properties), denial-of-service attacks on sender or recipient (e.g., due to putting the ASN entry "out-of-sync" on either end), or frame counter reuse (since 802.15.4e-2012 does not inspect the frame counter in the device descriptor, but solely relies on the ASN entry). Thus, ASN entries are very fragile and their use should happen with extreme care.

7. As already mentioned, ASN anomalies may seriously impact security. If any device’s ASN state is out-of-synch with other devices, this may result in that device not being able to communicate in the network any more. With network-wide keys, the remedy may include a combination of rekeying all devices (a costly proposition) and resetting ASN entries of the impacted device.

8. The security provisions in 802.15.4-2011 and 802.15.4e-2012 leave some room for potential Denial of Service (DoS) attacks. We only discuss "accidental" DoS attacks for now, which we define as

those triggered without active involvement of an adversarial network element (active DoS attacks are considered separately).

* If a device acts on an incoming frame that is cryptographically secured, it has assurances that this frame originated from a device with access to the key. Here, processing a frame with a key provides a mechanism for network segregation, since proper incoming security processing (and assuming non-compromised locally stored security-relevant material and processes) allows one to draw conclusions as to whether originator and recipient belong to the same "group" (the key-sharing group). This propery holds if the incoming frame has an authenticity tag; in some cases, this may also hold if the frame was only encrypted, but not authenticated. This "network segregation" property holds independent of whether the key was actually a real key (cryptographic key); the number of groups created depends on the number of these group keys (perhaps, more properly termed "group identifiers" if of no cryptographic use) used.

* A joining node must make its decision to join the network based on information derived from processing an enhanced beacon. Since it is in embryonic stage, it has to take this information at face value (no matter whether this beacon was cryptographically secured or not). In theory, this may give rise to dilemmas of choice, i.e., how is a joining node to pick which beacon to act upon? As already said, one could realize network segregation using a "default" key, whereby the joining node and the beaconing device would be able to check membership of the same loosely defined group (this is the mechanism Wireless HART uses). However, as mentioned before, this could potentially adversely impact 802.15.4-2011 and 802.15.4e-2012 security. Even if one discards security concerns, this only establishes membership of a very crudely defined group (e.g., if one uses as "default" key the fixed value "6tisch-default-join", this would have any joining node accept any 6tisch-beacon). The same filtering mechanism could also, without any possible security side effects, be realized by partitioning the "language of well-formed frames" and, e.g., filtering enhanced beacons on the data object "6tisch-default-join" (e.g., when including this tag as a Header Information Element with the beacon). If one does not use such explicit "tags", one could conceivable also accept beacon frames that implement an alien protocol, rather than 802.15.4e-2012. It is, however, quite unlikely that a random alien frame will pass incoming frame filtering, since 802.15.4 incoming frame processing checks for well-formedness. Checking some built-in redundancy of well-formed frames

thereby most likely filters out virtually all unwanted alien frame types. Such filtering could, e.g., include a "language check" as to fixed fields in information elements. For enhanced beacon frames for TSCH, e.g., the header fields of the synchronization IE, timeslot IE, and header IE contained herein have fixed 2-octet values 0x1a06, 0x1c01, and 0x1d01, respectively, thereby providing up to 48 bits of redundancy. This provides similar filtering functionality as the explicit "6tisch-default-join" tag mentioned before, but without the need to introduce an explicit tag or to communicate this separately over the air.

It should be emphasized (again) that none of the mechanisms above protects against active attacks.

1.3. Join Protocol Behavior

1.3.1. Device Enrollment Phases

The join protocol consists of three phases, viz.

1. Device Authentication: The joining node and proxy network node authenticate each other and establish a shared key, so as to ensure on-going authenticated communications. This may involve a server as a third party.

2. Authorization: The proxy network node decides on whether/how to authorize a joining node (if denied, this may result in loss of bandwidth). Authorization decisions may involve other nodes in the network.

3. Configuration/Parameterization: The proxy network node distributes configuration information to the joined node, such as scheduling information, IP address assignment information, and network policies. This may originate from other network devices, for which it acts as proxy. This step may also include distribution of information from the joining node to the network node and other nodes in the network and, more generally, synchronization of information between these 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 keys of their 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. Mutual authentication is performed between the joining node and the proxy using their certificates, which also results in a shared key between these two

entities. The proxy assists the joining node in mutual authentication with the server, which also results in a shared (end-to-end) key between those two entities. The server may arbitrage network authorization of the joining node (where the proxy will deny bandwidth if authorization is not successful) and may distribute network-specific configuration parameters (including network-wide keys) to the joining node. In its turn, the joining node may provide distribute/synchronize information (including, e.g., network statistics) to the server node.

The server functionality is a role and may be implemented with one device (centralized) or with multiple entities (distributed). In either case, mutual authentication is established with each physical server entity with which a role is implemented. Note that in the above description, the proxy does not solely act as a relay node. For more detailed rationale, see the relevant detailed descriptions further in this document. This also provides some insight into what happens in case the initial set-up requirements are not met or some other out-of-sync behavior occurs and suggest some optimization in case server-related information is already available with the proxy node (caching).

When a device rejoins the network in the same authorization domain, the authorization step could be omitted if the server distributes the authorization state for the device to the proxys when the device initially joined the network. However, this generally still requires the exchange of updated configuration information, e.g., related to time schedules and bandwidth allocation.

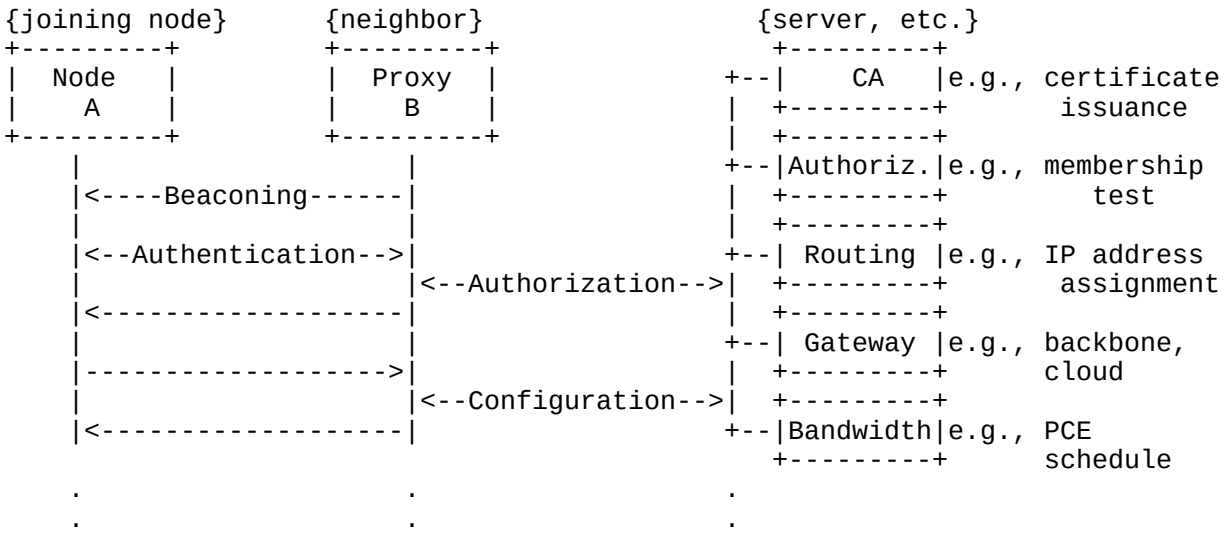


Figure 1: Network joining, with only authorization by third party

1.3.2. Join protocol description

NOTE: the description below considers the scenario where devices have credentials on board and where the neighbor does not simply act as a relay node only. Other scenarios will be considered in future versions of this draft.

1. Upon hearing the enhanced beacon, the joining node stores the PAN descriptor.

2. The joining node uses local criteria, including information contained in the PAN desciptor, to determine whether it wishes to join the network.

3. The joining node sends the first join protocol message to the neighbor node. This message corresponds to one or more unsecured MAC data frames. This message includes the joining node’s key contribution and credentials.

4. The neighbor node processes the incoming join message from the joining node and, depending on local criteria (including a check that this is a join message), grants the joining node temporary diplomatic immunity status ("exempt stauts") from a MAC perspective (if not granted, this simply results in a rejected incoming frame at the MAC layer).

5. The neighbor node performs some checks on the incoming message. If successful, it sends a first return join protocol message to the joining node. This message corresponds to one or more unsecured MAC data frames. This message includes the neighbor node’s key contribution and credentials. It may also include the server’s cached first return join protocol message info. At this point, the neighbor node is capable of deriving the shared key with the joining node based on inputs received and locally maintained status information.

6. The joining node performs some checks on the incoming message (including that it received this message from the neighbor node and that this is a join message). If succesful, it derives a shared key with the neighbor node and may derive a shared key with the server (it may also postpone the latter till required ["lazy evaluation"]).

7. The joining node sends a second join protocol message (a key confirmation message) to the neighbor node and may include some other information (so-called piggy-backed info). The piggy-backed information includes configuration information to be passed from the joining node to the neighbor node. This message corresponds to one or more unsecured MAC data frames.

8. The joining node sends a similar second join protocol message (another key confirmation message, including piggy-backed information) to the server. The piggy-backed information includes configuration information to be passed from the joining node to the server that allows the server to check the joining node’s true credentials and some network-relevant parameters (including the ASN number and the joining node’s local schedule maintained with the neighbor node). This message corresponds to one or more unsecured MAC data frames. This message may be combined with the message sent to the neighor node, since it travels along the same initial communication path.

9. The neighbor node checks the received second join protocol message (the key confirmation message and received piggy-backed info), including that this message originated from the same device as the previous join protocol message and that this message is a join message. If successful, it clears the "exempt status" attribute of the joining node in the DeviceDescriptor (thereby, lifting diplomatic immunity status for the joining device) and adds the {data key, joining node} pair to its KeyDescriptor list. It also stores policy-related attributes for this key. It may update some additional state, based on the piggy-backed info received from the joining device. The clearing of the "exempt status" flag means that it will only

accept incoming secured frames from the joining node from that moment onwards.

10. The server checks the received second join protocol message (key confirmation message and received piggy-backed info). If successful, it adds the {data key, joining node} pair to its locally maintained list of end-to-end keying material and includes policy-related attributes for this key. It sends its own second return join protocol message (another key confirmation message, including piggy-backed configuration information) to the joining node. This is actually sent to the neighbor node it received the first join protocol message from, who in turn forwards this to the joining node (here, the neighbor node acts in storing mode and knows the local network topology the server may not know (yet)). NOTE: this requires the neighbor node to remember some information pertaining to the joining node (mainly, the {data key, joining node} pair of the KeyDescriptor and the local communication schedule with the joining node). This may include an explicit notification to the neighbor node that the joining node is authorized to join the network. If so, this authorization part of this message is secured, using end-to-end security between the server and the neighbor node.

11. The neighbor node checks the authorization-related info, if indeed contained in this message (if denied, it may clear the joining node related info from its tables). If successful, it forwards this information along with its own second return join protocol message (key confirmation message and piggy-backed info) to the joining node. Obviously, this can be done separately as well, but travels over the same (single hop) communication path.

12. The joining node checks the received second join protocol message (the key confirmation and piggy-backed info) from its neighbor node. If successful, it adds the {data key, neighbor node} pair to its KeyDescriptor list. It also stores policy-related attributes for this key. If not successful, it clears its local table with info pertaining to the neighbor node.

13. The joining node checks the received second join protocol message (the key confirmation and piggy-backed info) from the server. If successful, it adds the {data key, server node} pair to its locally maintained list of end-to-end keying material and includes policy-related attributes for this. It may also update its local state, based on information contained in the piggy-backed info received from the server. Updates of local state may be subject to additional local criteria, such as consistency

of status information obtained from neighbor node and server node (e.g., pertaining to the ASN field, PAN identifier, or scheduling information). This may give rise to triggered alerts. If not successful, it clears its local table with info pertaining to the server node. Depending on local criteria, it may clear the table with info pertaining to the neighbor node.

1.3.3. Remarks

1. The join protocol above can be optimized in various ways, including first handling mutual authentication of local communication channels, prior to engaging in non-local communications so as to reduce time latencies in case of failure conditions. This is realized by having the neighbor node authenticate itself to the joining node before initiating non-local communications from the joining node to the server node along the communication path via the neighbor node (rather than at the end of this non-local communications). Since 10-hop communications may take roughly 2.5 minutes on a TSCH network and local communication time latencies take roughly 15 seconds, this could present a significant time saving (and reduced requirement on keeping state and energy consumption on the joining device).

2. The join protocol above takes only one non-local communication between the neighbor node and the server node. This assumes that the neighbor node is able to cache security-related information from the server. Since this includes certificate-related information of the server node (which may require more than one classical 802.15.4 MAC frame to carry), this may present significant communication time latency savings. Obviously, an additional long-haul round trip may be required should this cached information be stale (keeping this information in sync is a responsibility of the neighbor node). With caching, this turns the join protocol described above into the most efficient possible, in terms of communication time latencies involved. At the same time, this protocol has very strong security properties, unmatched by legacy protocols [...].

3. The join protocol above assumes authentication of the joining node to the neighbor node, before non-local traffic takes place. This assists in thwarting denial-of-service attacks on "das Hinterland" of the neighbor node triggered by joining nodes with improper credentials (unparsable certs). While this check is an authentication check only and *not* a fine-grained authorization check, this could be complemented by additional local "sanity checks" on the neighbor node (device white listing, etc.), thus allowing extensibility to more fine-grained authorization

filtering mechanisms. (Further details are outside scope of this document, but may be described later.)

4. The join protocol above assumes authentication of the neighbor node to the joining node (i.e., the neighbor node is not simply a relay node). This potentially assists in thwarting denial of service attacks on the joining node itself, primarily since it may allow the joining node to conclude it joined an improper network based on local communications only (if the neighbor node presented an unparsable cert or did not properly authenticate), rather than having to await a nonlocal verdict via the server that may take a long time to materialize. Here, again, more fine-grained authorization checks may be realized in scenarios where the joining node has more local intelligence to draw from. (Again, further details are outside scope of this document for now.)

5. The join protocol above includes mutual authentication between the joining node and the neighbor node and establishment of a shared "link key" (to use 802.15.4 parlance) between these two devices. This may be useful in case one wishes to trigger time synchronization between the joining node and the neighbor node contingent on frames secured using this pair-wise key only. This would strengthen TSCH security compared to that provided by the current 802.15.4e-2012 specification (which allows time synchronization to be also triggered by frames secured using a network-wide key, thereby opening the network to attacks by a single random compromised node, rather than a specific compromised node [the "clock tower" node] only.)

6. The join protocol above can also be "weakened", e.g., by removing authentication of the neighbor node to the joining node or vice-versa. As already said, this might open the protocol to wide-spread denial of service attacks on the network (in case the neighbor node simply forwards any joining node traffic, without inspection) or denial of service attacks on the joining node (in case the neighbor node is a bogus node or a node of an alien network). In some settings, though, practical trade-offs may favor such a "weakened" approach, e.g., if one wishes to "sprinkle" in sufficiently many neighbor nodes to guarantee connectivity to the server during initial deployment. If so, one should still have a fall-back strategy in place should denial of service attacks become a reality. (NOTE: These "weakened" versions will be analyzed in more detail in a later version of this draft.)

7. The join protocol above does not impose requirements on the security of the communication path between the neighbor node and

the server, except that "it should be there" (i.e., there is connectivity) although there may be additional requirements to counter, e.g., denial of service attacks on communications between neighbor node and server. (An exception here is if the server returns authorization-related information to the neighbor node [which we required to be secured], but which we will ignore for now.) Such minimization of dependencies between the join protocol and the routing protocol may be beneficial for use cases where one wishes to facilitate "random" installation process flows. Obviously, once a node is part of the network, it should be able to route packets (but that is not part of the join protocol itself, but next-stage phase).

8. The join protocol above tries to embrace a design where the order of joining would be mostly orthogonal to routing protocol topology considerations, if it all possible. In particular, it is aimed to take into account that not all installations follow the pattern where one has an operational network and where all non-local communications during the join protocol not of the type {joining node - neighbor router} are within the operational network (i.e., one would like to facilitate scenarios other than a tree-like structure, where network is built from tree root up onwards [this is highly relevant in building control settings]).

9. The join protocol above exchanges piggy-backed information between joining node, neighbor node, and server. This conceptually would allow very agressive implementations of the routing protocol, where one intertwines routing and join processes, by including some of the routing-related attributes as opaque strings in the piggy-backed fields. It should be noted that the join protocol already supports the routing tree of the existing network and the "new tree branch" {joining node - neighbor node}, so all "upwards routes" to the pre-existing tree roots are inherited right away. The only routes that may need defining are those towards the newbee joining node. For reliability reasons, this does require the joining node to have successfully concluded the join protocol first. As such, there seems to be no technical reason to intertwine these protocols: one should simply perform routing-related operations only *after* the join protocol ran its full course.

10. The join protocol above allows the neighbor node to influence with which server the joining node communicates, thus allowing a distributed implementation of the server.

11. The join protocol above assumes that the server arbitrages the correct value of supposedly common network parameters, such as the PAN identifier and ASN field. Here, one should note that

the neighbor node can indicate, e.g., any PAN identifier and any ASN entry to its liking in its beacon, which does not necessarily correspond to the "common world view" hereof by the server.

12. The join protocol above could in theory result in a node joining the network only locally (i.e., forming a two-node network with the neighbor node only), without the server or any other nodes becoming aware of this. This scenario could arrise if the joining node is unaware of some server-related context information and if the neighbor node simply usurps the server role itself. The impact of this "hidden node" type scenario depends on higher-layer, end-to-end design details. From a MAC perspective, this could simply mean that the two-node {joining node, neighbor node} network is conceptually represented by this neighbor node, where the internal structure of this two-node network remains hidden for other nodes.

1.4. Routing Behavior

TBD.

2. IANA Considerations

There is no IANA action required for this document.

3. Acknowledgments

TBD. Kris Pister provided the filtering example details for enhanced beacon frames. Yoshi Ohba, Subir Das, Giuseppe Piro, Pascal Thubert, and Kris Pister kindly provided feeback on previous versions of this document.

4. Normative References

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

[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-02 (work in progress), July 2014.

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

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/.

Copyright Notice

Table of Contents

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 generaic 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

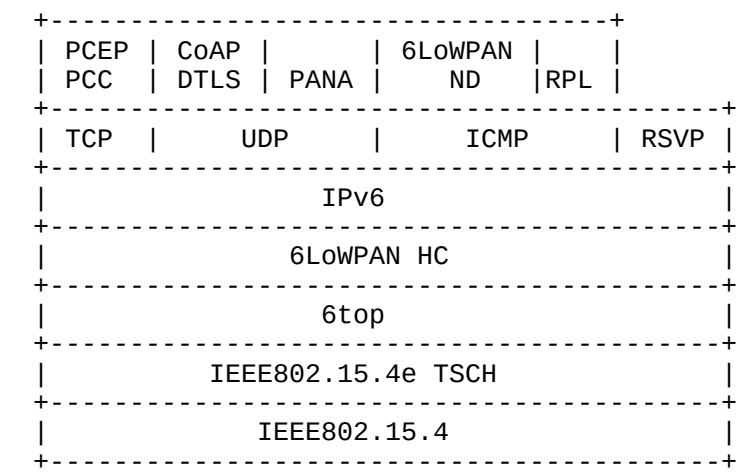


Figure 1

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

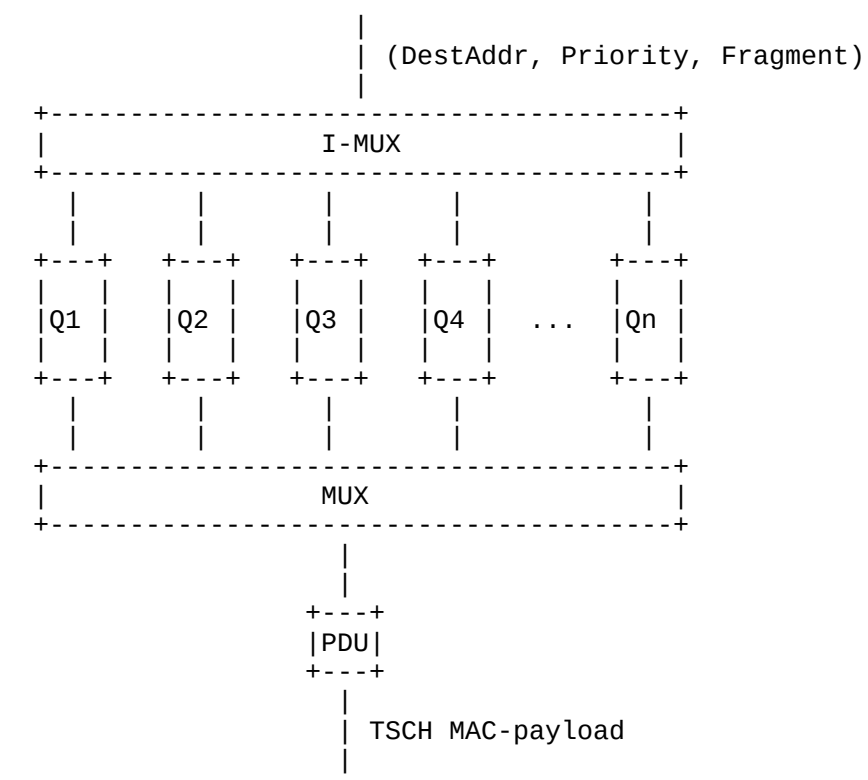


Figure 2

In Figure 2, Qi 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 (StatisticsMetriceID and StatisticsValue)

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

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 exits.

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 exits.

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 decribed 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 exists. 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 exists.

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 an 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 an 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 an 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 an 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 channalOffset 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 channalOffset 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).

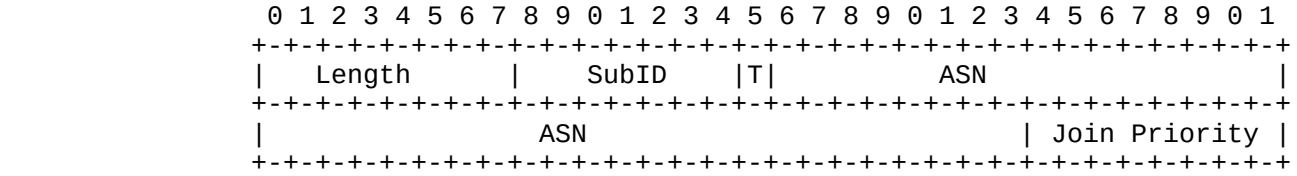


Figure 3

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).

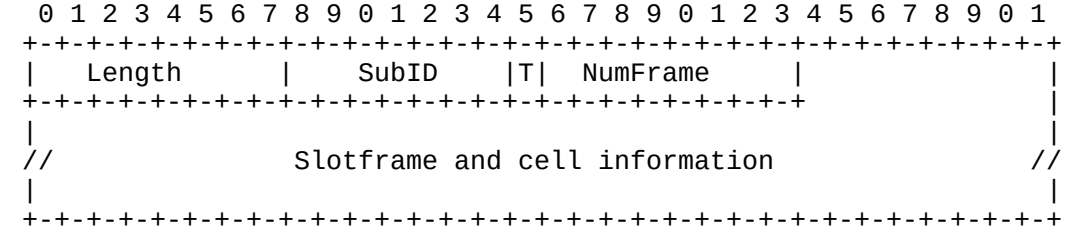


Figure 4

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.

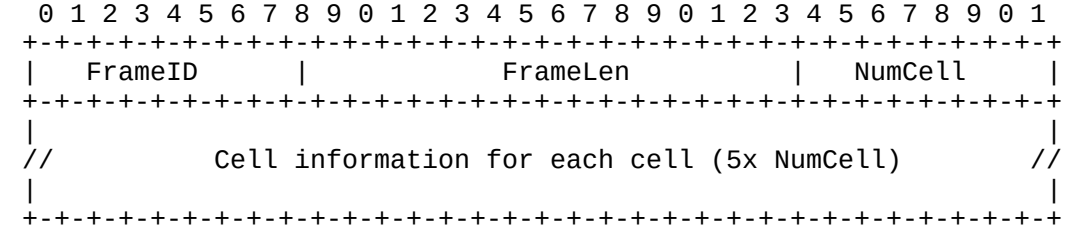


Figure 5

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.

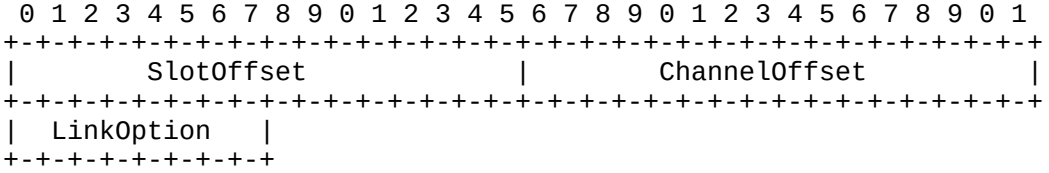


Figure 6

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).

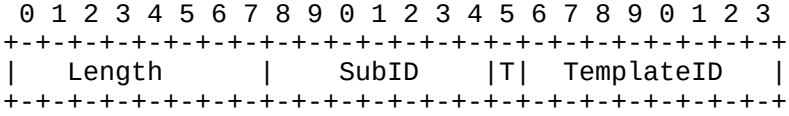


Figure 7

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).

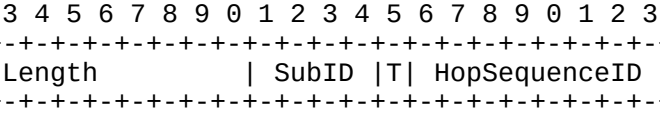


Figure 8

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).

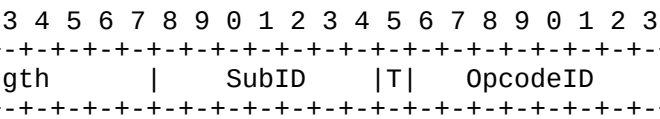


Figure 9

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).

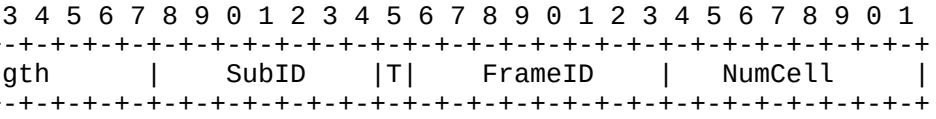


Figure 10

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 OpecodeID of Reserve Soft Cell Request, NumCell presents how many cells are required to reserve; and when BwIE is combined with the OpecodeID 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).

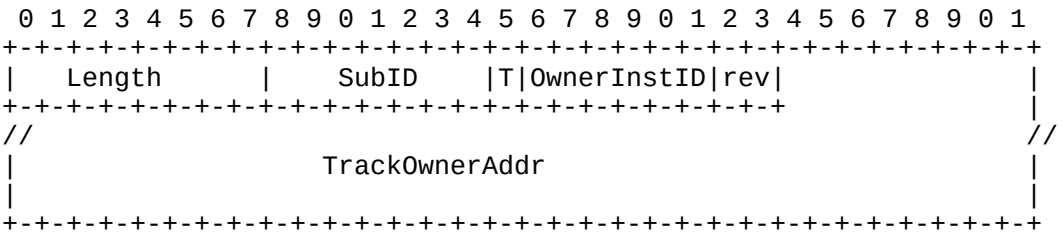


Figure 11

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).

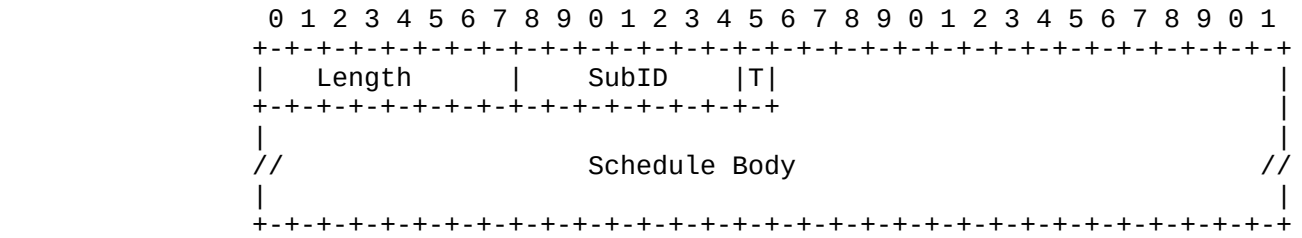


Figure 12

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

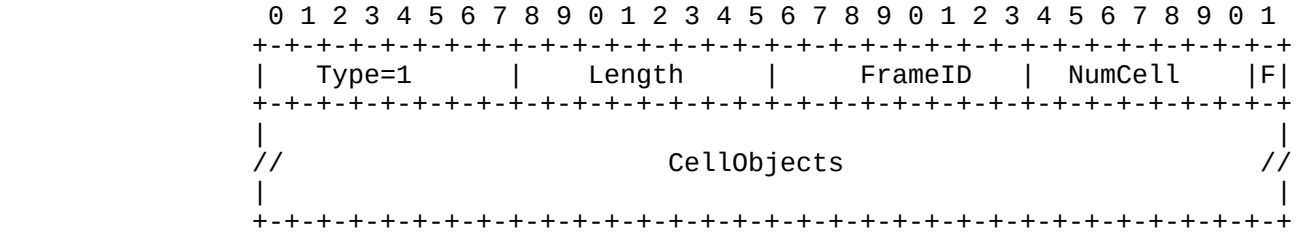


Figure 13

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

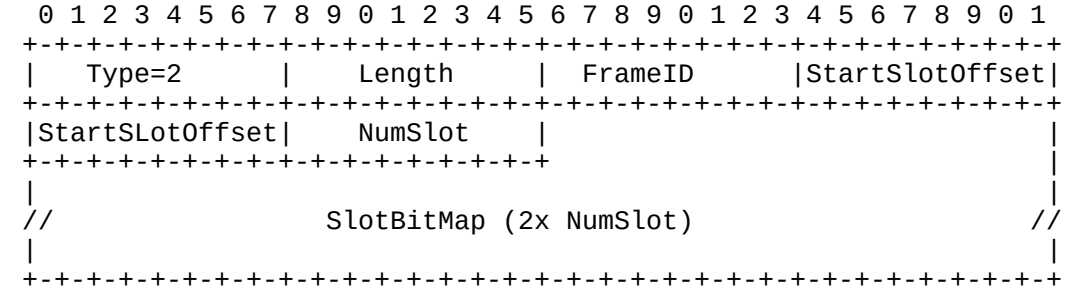


Figure 14

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

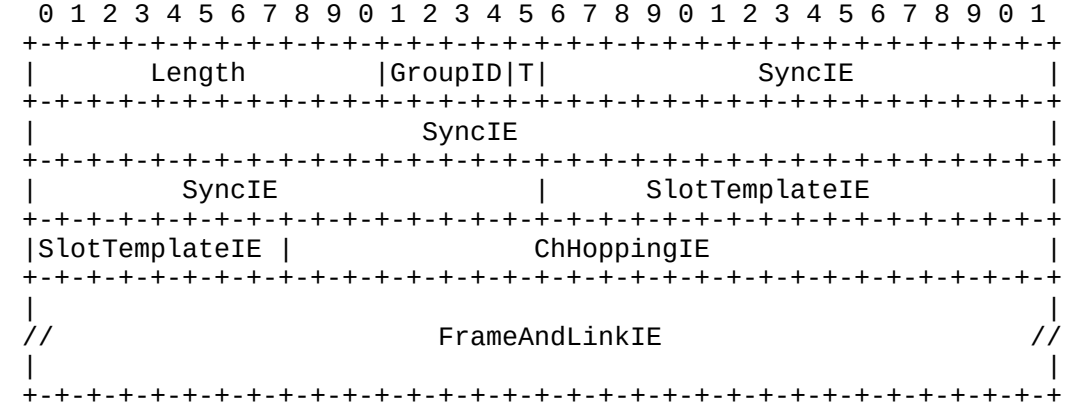


Figure 15

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

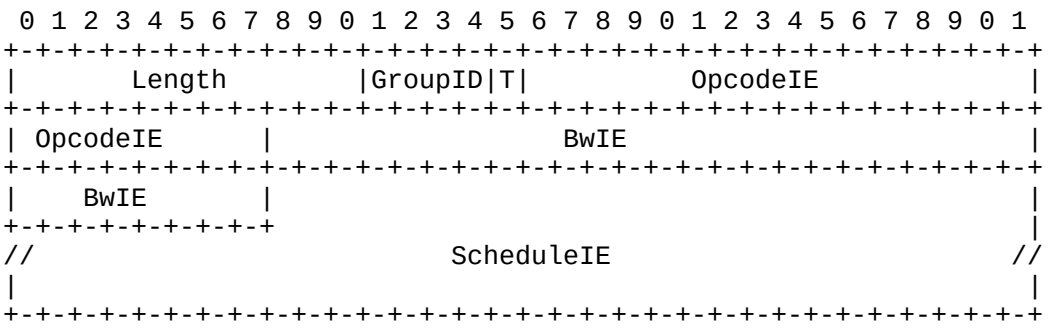


Figure 16

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

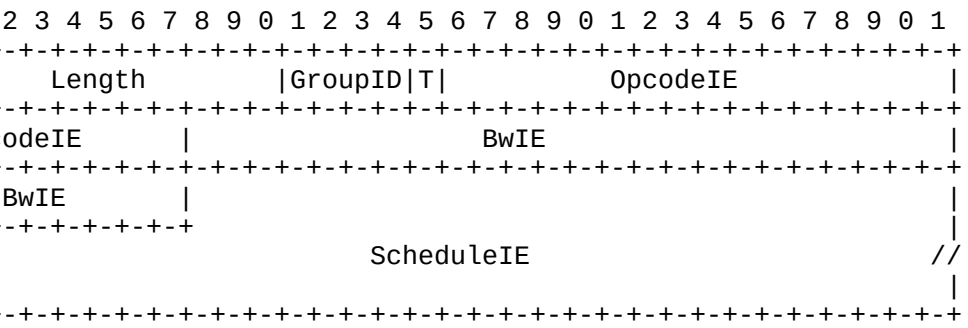


Figure 17

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

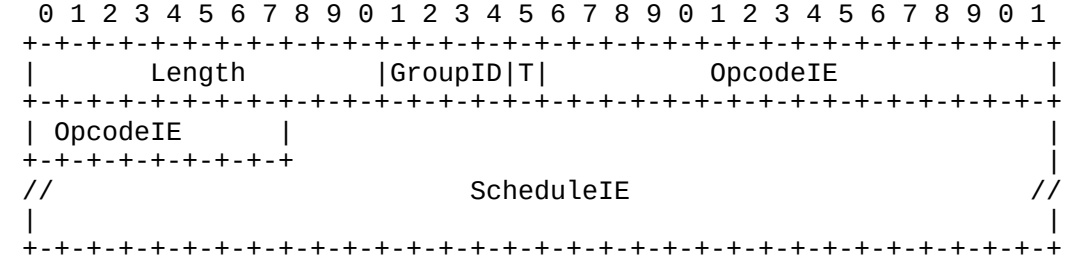


Figure 18

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

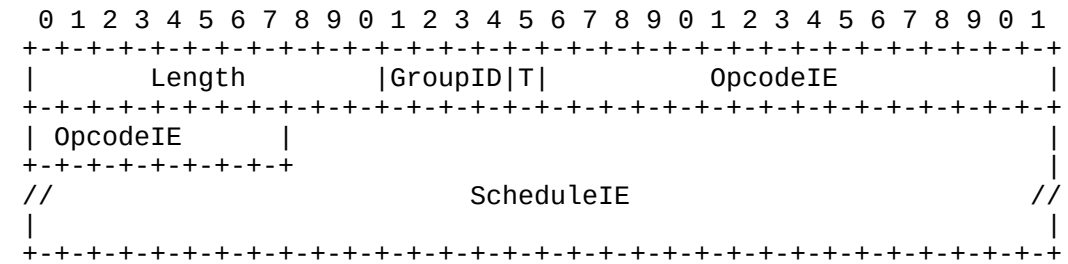


Figure 19

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

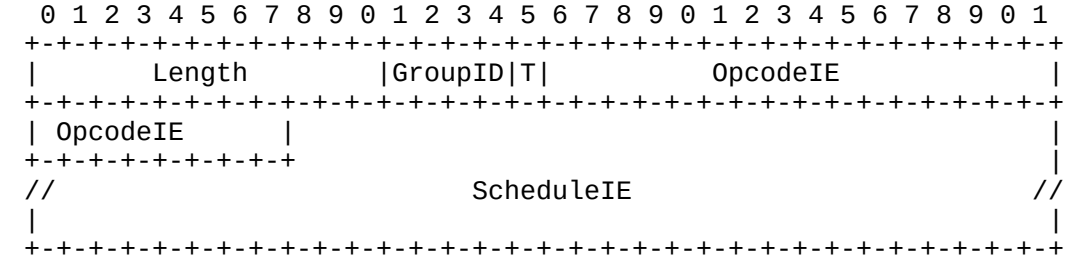


Figure 20

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: SUM(NumOfCell(i)*NumOfBytePerPacket)/(FrameLen(i)*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 trggiers 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

6TiSCH
Internet-Draft
Intended status: Standards Track
Expires: September 7, 2015

Z. Chen
C. Wang
InterDigital Communications, LLC

March 6, 2015

Use Cases and Requirements for using Track in 6TiSCH Networks
draft-wang-6tisch-track-use-cases-00

Abstract

This document further analyzes the 6TiSCH requirements related to Track through the use of examples and use cases. The goal of this document is to trigger discussions in 6TiSCH working group so that all relevant considerations are take into account when design Track reservation schemes in 6TiSCH.

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 September 7, 2015.

Copyright Notice

Table of Contents

1. Introduction

IEEE802.15.4e [IEEE802154e] was published in 2012 as an amendment to the Medium Access Control (MAC) protocol defined by the IEEE802.15.4-2011 [IEEE802154] standard. IEEE802.15.4e will be rolled into the next revision of IEEE802.15.4, scheduled to be published in 2015. The Timeslotted Channel Hopping (TSCH) mode of IEEE802.15.4e is the object of this document. The 6TiSCH working group is chartered to enable IPv6 over the TSCH mode of the IEEE802.15.4e standard.

The requirements for 6TiSCH are well documented [I-D.ietf-6tisch-tsch]. Initially, the WG will limit its scope to distributed routing over a static schedule. In this draft, we focus and expand discussions pertaining to Track. We propose requirements and use cases for different type of Track reservation schemes.

2. Terms used in this document

The draft uses terminologies defined in [I-D.ietf-6tisch-terminology]. The following are definition of terminologies used in this draft.

Centralized Track reservation: The reservation of a track done by the central controller of the network, e.g. PCE.

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

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

3. Use Cases: Industrial Networks

An industry network is a good use case for a 6TiSCH network. In an industry network as shown in Figure 1, many devices are LLN devices, e.g. sensors and actuators. There are many types of applications in an industry network, such as industry process control and automation applications, e.g. an automation assembly line, and industry monitor applications, e.g. a safety monitoring application.

3.1. Industry process control and automation applications

In an industry process control and automation application as shown in Figure 1, LLN Devices are actuator and sensors in an automation assemble line. An LLN Device, for example LLN Device 1, MAY periodically send signalling packets to another actuator, e.g. LLN Device 2. For example, LLN Device 1 locate at the step 1 of the automation assemble line, whenever it finishes a task, it will send singling packets to LLN Device 2 located at the step 2 of the automation assemble line to trigger the next action in the automation assembly line. The delay of these packets are extremely important for the performance of the automation assembly line. Also the reliability of these signalling packets are extremely important since a packet loss may result products with defects. Reserving a Track between LLN device 1 and LLN device 2 can not only guarantee the delay of these signalling packets but also improve the reliability of these packet due to less interference. Moreover, by reserving a Track, battery powered LLN Devices are able to wake up and sleep based on its TSCH schedule to save energy. In these cases, the Tracks reserved are deterministic, unless the topology of the network changes.

3.2. Industrial monitoring applications

In an industrial monitoring application, sensors such as LLN 1 and 2, monitor the status of each machine or plant and send data to the Control Controller as shown in Figure 1. An LLN Device, for example LLN Device 1, MAY detect a critical event, and sends a signalling emergency message to the Central Controller in the network. After that the LLN Device may send monitoring data to the Central Controller. The singling packets that contains an emergency message SHOULD arrive at the Central Controller with minimum delay and highest reliability. Therefore, multiple Tacks may be reserved between these sensors and the Central Controller. Moreover, a bursty traffic that contains monitoring data MAY follow the critical message. These data packets also require low latency and high reliability, thus a high bandwidth Track SHOULD be quickly set-up between these LLN Devices and the Central Controller. Therefore, the Track reservation scheme has to react faster in a more dynamic way.

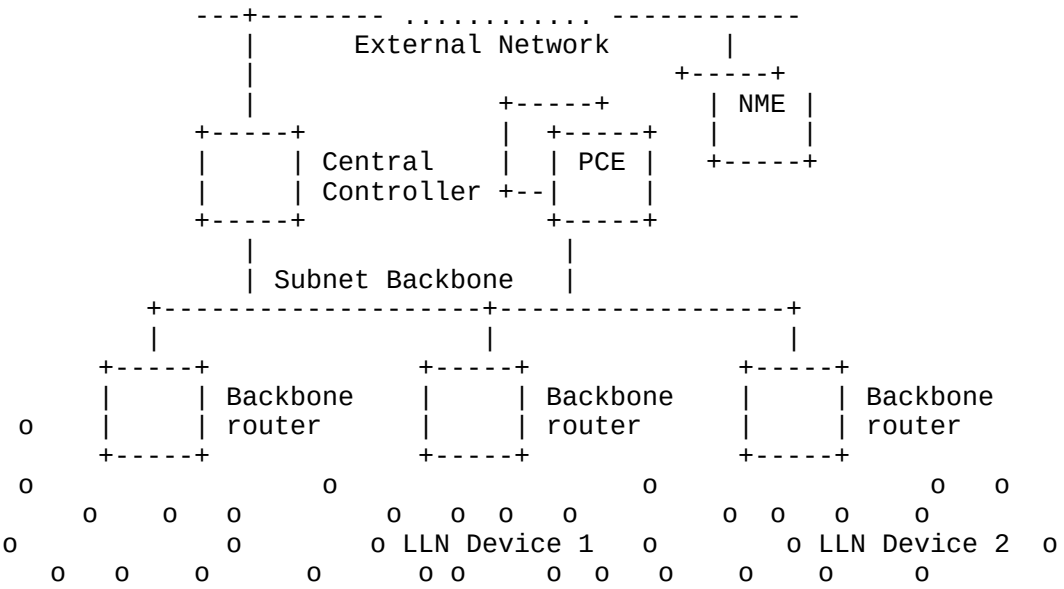


Figure 1: Use Case of an Industry Network

4. Handling Tracks in 6TiSCH Networks

4.1. General Behavior of Tracks

In this section, we discuss the behavior and the benefits of Tracks. As discussed in [I-D.ietf-6tisch-architecture], Track is first a multi-hop paths from the source LLN Device to the destination LLN Device. Second, some resources of LLN Devices on the path are reserved by configuring their TSCH schedule. Therefore, an LLN Device on the Track not only knows what cells it should use to receive packets from its previous hop, but also knows what cells it should use to send packets to its next hop. There are several benefits for using Track to forward a packet from the source LLN Device to the destination LLN Device.

First, Track forwarding as described in Section 10.1 in [I-D.ietf-6tisch-architecture] is a layer-2 forwarding scheme, which introduces less process delay and overhead than layer-3 forwarding scheme. Therefore, LLN Devices can save more energy and resource, which is critical for resource constrained devices.

Second, since channel resources, i.e. cells, have been reserved for communications between LLN devices of each hop on the Track, the packets traverse along the Track as a train passes each stations along the rail track. Therefore, the throughput and delay of the traffic on a Track is guaranteed and the jitter of the traffic is small. These are extremely important features for time-sensitive applications, which require packets arrives on time.

Third, by knowing the scheduled time slots of incoming cell and outgoing cell, LLN devices on a Track could save more energy by staying in sleep state during in-active slots. This is extreme important for LLN Devices that are battery powered.

Fourth, by allocating scheduled channel frequency, both inter-Track and intra-Track interference can be reduced. This will enhance the reliability of transmissions on a Track and reduce energy consumption of LLN Devices by decreasing the number of retransmissions.

4.2. Track Reservation

Cells along a Track have to be reserved before any packet transmissions. How to efficiently allocate resources along a Track becomes a challenging problem. Generally, there are both remote Track management and hop-by-hop Track management described in [I-D.ietf-6tisch-architecture] to solve the Track reservation issue.

4.2.1. Remote Track Management

In the remote Track management scheme in section 9.3 in [I-D.ietf-6tisch-architecture], a central controller of the network, e.g. Path Computation Element (PCE) in Figure 1, can allocate hard cells of LLN Devices on a Track remotely. The network may be globally optimized by the central controller of the network.

4.2.2. Hop-by-hop Track Management

In the hop-by-hop Track management scheme in section 9.4 in [I-D.ietf-6tisch-architecture], LLN Devices can negotiate and reserve Soft Cells in their TSCH Schedule by communicating with each other. By configuring the TSCH Schedule of LLN Devices on a route, a Track can be reserved to enhance the multi-hop communications between the source and the destination. The hop-by-hop Track management schemes may be more scalable and robust than the remote Track management scheme since it does not rely on the central controller of the network.

5. Requirement for Track reservation schemes

The track reservation schemes are required to support both deterministic traffics such as periodical transmissions for industry process control and automation applications and dynamic traffics such as bursty transmissions for industrial monitoring applications.

5.1. Centralized Track reservation

Need a protocol for LLN devices to report their topology and TSCH schedule information to the central controller as shown in Figure 1. The central controller need the topology information to obtain a path from the source to the destination and the network can be better optimized if the central controller is aware of the TSCH schedule of all or part of LLN Devices in the network.

Need a lightweight protocol for the central controller to configure hard cells of LLN Devices using 6top interface defined in [I-D.ietf-6tisch-6top-interface]. The central controller has to configure hard cells of LLN Devices on the track remotely and LLN Devices are usually constrained devices which may not support heavyweight protocol such as RFC 5440 [RFC5440]

5.2. Distributed Track reservation

Need a fast reaction protocol to reserve a Track. LLN Devices have limited information about the topology of the network and the TSCH schedule of other LLN Devices on the path. The protocol should

quickly detect a Track reservation failure. Need an efficient negotiation protocol between LLN Devices multi-hop away from each other. LLN Devices on the path have to negotiate in order to reserve a Track, which may bring extra overhead to constrained devices.

6. Conclusions

A Track can provide low latency, guaranteed throughput and high reliable for end-to-end communications. There are many use cases that can show the benefit of using a Track, such as industry networks, home networks, structure networks, health networks and vehicular networks. Moreover, different Track reservation schemes, such as centralized and distributed schemes, need to be proposed to handle a large variety of requirements.

7. Security Considerations

This draft discussed the design considerations and operations of using Track in 6TiSCH networks. It does not introduce new security threats.

8. IANA Considerations

This specification does not require IANA action.

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.

9.2. Informative References

[I-D.ietf-6tisch-6top-interface]
Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top) Interface", draft-ietf-6tisch-6top-interface-02 (work in progress), October 2014.

[I-D.ietf-6tisch-architecture]
Thubert, P., Watteyne, T., Struik, R., and M. Richardson, "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-architecture-05 (work in progress), January 2015.

[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-03 (work in progress), January 2015.

[I-D.ietf-6tisch-tsch]
Watteyne, T., Palattella, M., and L. Grieco, "Using IEEE802.15.4e TSCH in an IoT context: Overview, Problem Statement and Goals", draft-ietf-6tisch-tsch-05 (work in progress), January 2015.

[RFC5440] Vasseur, JP. and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.

9.3. External Informative References

[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.

[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

Zhuo Chen
InterDigital Communications, LLC
781 Third Ave
King of Prussia, PA 19406
USA

Phone: +1 610 878 5730
Email: Zhuo.Chen@InterDigital.com

Chonggang Wang
InterDigital Communications, LLC
781 Third Ave
King of Prussia, PA 19406
USA

Phone: +1 610 878 5831
Email: Chonggang.Wang@InterDigital.com

detnet
Internet-Draft
Intended status: Informational
Expires: April 30, 2015

P. Wetterwald
Cisco
J. Raymond
Hydro-Quebec

October 27, 2014

Deterministic Networking Uitilities requirements draft-wetterwald-detnet-utilities-reqs-01

Abstract

This paper documents the needs in Smart Grid industry to establish multi-hop paths for characterized flows with deterministic properties .

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 April 30, 2015.

Copyright Notice

Table of Contents

1. Introduction

[I-D.finn-detnet-problem-statement] defines the characteristics of a deterministic flow as a data communication flow with a bounded latency, extraordinarily low frame loss, and a very narrow jitter. This document intends to define the utility requirements for deterministic networking.

2. Requirements Language

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

3. Overview

Evolution of Utility Telecom Networks

The business and technology trends that are sweeping the utility industry will drastically transform the utility business from the way it has been for many decades. At the core of many of these changes is a drive to modernize the electrical grid with an integrated telecommunications infrastructure. However, interoperability, concerns, legacy networks, disparate tools, and stringent security requirements all add complexity to grid transformation. Given the range and diversity of the requirement that should be addressed by the next generation telecommunications infrastructure utilities need to adopt a holistic architectural approach to integrate the electrical grid with digital telecommunication across the entire power delivery chain.

Many utilities still rely on complex environments formed of multiple application-specific, proprietary networks. Information is siloed between operational areas. This prevents utility operations from realizing the operational efficiency benefits, visibility, and functional integration of operational information across grid applications and data networks. The key to modernizing grid telecommunications is to provide a common, adaptable, multi-service network infrastructure for the entire utility organization. Such a network serves as the platform for current capabilities while enabling future expansion of the network to accommodate new applications and services.

To meet this diverse set of requirements, both today and in the future, the next generation utility telecommunnication network will be based on open-standards-based IP architecture. An end-to-end IP architecture takes advantage of nearly three decades of IP technology development, facilitating interoperability across disparate networks and devices, as it has been already demonstrated in many mission-critical and highly secure networks.

IEC and different National Committees have mandated a specific adhoc group (AHG8) to define the migration strategy to IPv6 for all the IEC TC57 power automation standards. IPv6 is seen as the obvious future telecommunication technology for the Smart Grid. The Adhoc Group

will discose, to the IEC coordination group, their conclusions at the end of 2014.

It is imperative that utilities participate in standards development bodies to influence the development of future solutions and to benefit from shared experiences of other utilities and vendors.

4. Telecommunication Trends and General telecommunication Requirements

These general telecommunication requirements are over and above the specific requirements of the use cases that have been addressed so far. These include both current and future telecommunication related requirements that should be factored into the network architecture and design.

4.1. General Telecommunication Requirements

- IP Connectivity everywhere
- Monitoring services everywhere and from different remote centers
- Move services to a virtual data center
- Unify access to applications / information from the corporate network
- Unify services
- Unified Communications Solutions
- Mix of fiber and microwave technologies - obsolescence of SONET/ SDH or TDM
- Standardize grid telecommunication protocol to opened standard to ensure interoperability
- Reliable Telecommunications for Transmission and Distribution Substations
- IEEE 1588 time synchronization Client / Server Capabilities
- Integration of Multicast Design
- QoS Requirements Mapping
- Enable Future Network Expansion
- Substation Network Resilience

- Fast Convergence Design
- Scalable Headend Design
- Define Service Level Agreements (SLA) and Enable SLA Monitoring
- Integration of 3G/4G Technologies and future technologies
- Ethernet Connectivity for Station Bus Architecture
- Ethernet Connectivity for Process Bus Architecture
- Protection and teleprotection on IP

4.1.1. Migration to Packet-Switched Network

Throughout the world, utilities are increasingly planning for a future based on smart grid applications requiring advanced telecommunications systems. Many of these applications utilize packet connectivity for communicating information and control signals across the utility's Wide Area Network (WAN), made possible by technologies such as multiprotocol label switching (MPLS). The data that traverses the utility WAN includes:

- Grid monitoring, control, and protection data
- Non-control grid data (e.g. asset data for condition-based monitoring)
- Physical safety and security data (e.g. voice and video)
- Remote worker access to corporate applications (voice, maps, schematics, etc.)
- Field area network backhaul for smart metering, and distribution grid management
- Enterprise traffic (email, collaboration tools, business applications)

WANs support this wide variety of traffic to and from substations, the transmission and distribution grid, generation sites, between control centers, and between work locations and data centers. To maintain this rapidly expanding set of applications, many utilities are taking steps to evolve present time-division multiplexing (TDM)based and frame relay infrastructures to packet systems. Packet-based networks are designed to provide greater functionalities

and higher levels of service for applications, while continuing to deliver reliability and deterministic (real-time) traffic support.

4.2. Applications, Use cases and traffic patterns

Among the numerous applications and use cases that a utility deploys today, many rely on high availability and deterministic behaviour of the telecommunications networks. Protection use cases and generation control are the most demanding and can’t rely on a best effort approach.

4.2.1. Transmission use cases

Protection means not only the protection of the human operator but also the protection of the electric equipments and the preservation of the stability and frequency of the grid. If a default occurs on the transmission or the distribution of the electricity, important damages could occured to the human operator but also to very costly electrical equipments and perturb the grid leading to blackouts. The time and reliability requirements are very strong to avoid dramatic impacts to the electrical infrastructure.

4.2.1.1. Tele Protection

The key criteria for measuring Teleprotection performance are command transmission time, dependability and security. These criteria are defined by the IEC standard 60834 as follows:

- Transmission time (Speed): The time between the moment where state changes at the transmitter input and the moment of the corresponding change at the receiver output, including propagation time. Overall operating time for a Teleprotection system includes the time for initiating the command at the transmitting end, the propagation time over the telecommunications link and the selection and decision time at the receiving end, including any additional delay due to a noisy environment.

- Dependability: The ability to issue and receive valid commands in the presence of interference and/or noise, by minimizing the probability of missing command (PMC). Dependability targets are typically set for a specific bit error rate (BER) level.

- Security: The ability to prevent false tripping due to a noisy environment, by minimizing the probability of unwanted commands (PUC). Security targets are also set for a specific bit error rate (BER) level.

Additional key elements that may impact Teleprotection performance include bandwidth rate of the Teleprotection system and its resiliency or failure recovery capacity. Transmission time, bandwidth utilization and resiliency are directly linked to the telecommunications equipment and the connections that are used to transfer the commands between relays.

4.2.1.1.1. Latency Budget Consideration

Delay requirements for utility networks may vary depending upon a number of parameters, such as the specific protection equipment used. Most power line equipment can tolerate short circuits or faults for up to approximately five power cycles before sustaining irreversible damage or affecting other segments in the network. This translates to total fault clearance time of 100ms. As a safety precaution, however, actual operation time of protection systems is limited to 70- 80 percent of this period, including fault recognition time, command transmission time and line breaker switching time. Some system components, such as large electromechanical switches, require particularly long time to operate and take up the majority of the total clearance time, leaving only a 10ms window for the telecommunications part of the protection scheme, independent of the distance to travel. Given the sensitivity of the issue, new networks impose requirements that are even more stringent: IEC standard 61850 limits the transfer time for protection messages to 1/4 - 1/2 cycle or 4 - 8ms (for 60Hz lines) for the most critical messages

4.2.1.1.2. Asymetric delay

In addition to minimal transmission delay, a differential protection telecommunication channel must be synchronous, i.e., experiencing symmetrical channel delay in transmit and receive paths. This requires special attention in jitter-prone packet networks. While optimally Teleprotection systems should support zero asymmetric delay, typical relays can tolerate discrepancies of up to 750us.

The main tools available for lowering delay variation below this threshold are:

- A jitter buffer at the multiplexers on each end of the line can be used to offset delay variation by queuing sent and received packets. The length of the queues must balance the need to regulate the rate of transmission with the need to limit overall delay, as larger buffers result in increased latency. This is the old TDM traditional way to fulfill this requirement.

- Traffic management tools ensure that the Teleprotection signals receive the highest transmission priority and minimize the number

of jitter addition during the path. This is one way to meet the requirement in IP networks.

- Standard Packet-Based synchronization technologies, such as 1588-2008 Precision Time Protocol (PTP) and Synchronous Ethernet (Sync-E), can help maintain stable networks by keeping a highly accurate clock source on the different network devices involved.

4.2.1.1.2.1. Other traffic characteristics

- Redundancy: The existence in a system of more than one means of accomplishing a given function

- Recovery time : The duration of time within which a business process must be restored after any type of disruption in order to avoid unacceptable consequences associated with a break in business continuity.

- performance management : In networking, a management function defined for controlling and analyzing different parameters/metrics such as the throughput, error rate

- packet loss : One or more packets of data travelling across network fail to reach their destination

4.2.1.1.2.2. Teleprotection network requirements

The following table captures the main network requirements (this is based on IEC 61850 standard)

Teleprotection Requirement	Attribute
One way maximum delay	4-10 ms
Asymetric delay required	Yes
Maximum jitter	less than 250 us
Topology	Point to point, point to Multi-point
Availability	99.9999
precise timing required	Yes
Recovery time on node failure	less than 50ms - hitless
performance management	Yes, Mandatory
Redundancy	Yes
Packet loss	0.1% to 1%

Table 1: Teleprotection network requirements

4.2.1.2. Inter-Trip Protection scheme

Inter-tripping is the controlled tripping of a circuit breaker to complete the isolation of a circuit or piece of apparatus in concert with the tripping of other circuit breakers. The main use of such schemes is to ensure that protection at both ends of a faulted circuit will operate to isolate the equipment concerned. Inter-tripping schemes use signaling to convey a trip command to remote circuit breakers to isolate circuits.

Inter-Trip protection Requirement	Attribute
One way maximum delay	5 ms
Asymetric delay required	No
Maximum jitter	Not critical
Topology	Point to point, point to Multi-point
Bandwidth	64 Kbps
Availability	99.9999
precise timing required	Yes
Recovery time on node failure	less than 50ms - hitless
performance management	Yes, Mandatory
Redundancy	Yes
Packet loss	0.1%

Table 2: Inter-Trip protection network requirements

4.2.1.3. Current Differential Protection Scheme

Current differential protection is commonly used for line protection, and is typical for protecting parallel circuits. A main advantage for differential protection is that, compared to overcurrent protection, it allows only the faulted circuit to be de-energized in case of a fault. At both end of the lines, the current is measured by the differential relays, and based on Kirchhoff’s law, both relays will trip the circuit breaker if the current going into the line does not equal the current going out of the line. This type of protection scheme assumes some form of communication being present between the relays at both end of the line, to allow both relays to compare measured current values. A fault in line 1 will cause overcurrent to be flowing in both lines, but because the current in line 2 is a through following current, this current is measured equal at both ends of the line, therefore the differential relays on line 2 will not trip line 2. Line 1 will be tripped, as the relays will not measure the same currents at both ends of the line. Line

differential protection schemes assume a very low telecommunications delay between both relays, often as low as 5ms. Moreover, as those systems are often not time-synchronized, they also assume symmetric telecommunications paths with constant delay, which allows comparing current measurement values taken at the exact same time.

Current Differential protection Requirement	Attribute
One way maximum delay	5 ms
Asymetric delay Required	Yes
Maximum jitter	less than 250 us
Topology	Point to point, point to Multi-point
Bandwidth	64 Kbps
Availability	99.9999
precise timing required	Yes
Recovery time on node failure	less than 50ms - hitless
performance management	Yes, Mandatory
Redundancy	Yes
Packet loss	0.1%

Table 3: Current Differential Protection requirements

4.2.1.4. Distance Protection Scheme

Distance (Impedance Relay) protection scheme is based on voltage and current measurements. A fault on a circuit will generally create a sag in the voltage level. If the ratio of voltage to current measured at the protection relay terminals, which equates to an impedance element, falls within a set threshold the circuit breaker will operate. The operating characteristics of this protection are based on the line characteristics. This means that when a fault appears on the line, the impedance setting in the relay is compared to the apparent impedance of the line from the relay terminals to the fault. If the relay setting is determined to be below the apparent

impedance it is determined that the fault is within the zone of protection. When the transmission line length is under a minimum length, distance protection becomes more difficult to coordinate. In these instances the best choice of protection is current differential protection.

Distance protection Requirement	Attribute
One way maximum delay	5 ms
Asymetric delay Required	No
Maximum jitter	Not critical
Topology	Point to point, point to Multi-point
Bandwidth	64 Kbps
Availability	99.9999
precise timing required	Yes
Recovery time on node failure	less than 50ms - hitless
performance management	Yes, Mandatory
Redundancy	Yes
Packet loss	0.1%

Table 4: Distance Protection requirements

4.2.1.5. Inter-Substation Protection Signaling

This use case describes the exchange of Sampled Value and/or GOOSE message between IEDs in two substations for protection and tripping coordination. The two IEDs are in a master-slave mode.

The CT/VT in one substation sends the sampled analog voltage or current value to the Merging Unit (MU) over hard wire. The merging unit sends the time-synchronized 61850-9-2 sampled values to the slave IED. The slave IED forwards the information to the Master IED in the other substation. The master IED makes the determination (for example based on sampled value differentials) to send a trip command

to the originating IED. Once the slave IED/Relay receives the GOOSE trip for breaker tripping, it opens the breaker. It then sends a confirmation message back to the master. All data exchanges between IEDs are either through Sampled Value and/or GOOSE messages.

Inter-Substation protection Requirement	Attribute
One way maximum delay	5 ms
Asymetric delay Required	No
Maximum jitter	Not critical
Topology	Point to point, point to Multi-point
Bandwidth	64 Kbps
Availability	99.9999
precise timing required	Yes
Recovery time on node failure	less than 50ms - hitless
performance management	Yes, Mandatory
Redundancy	Yes
Packet loss	1%

Table 5: Inter-Substation Protection requirements

4.2.1.6. Intra-Substation Process Bus Communication

This use case describes the data flow from the CT/VT to the IEDs in the substation via the merging unit (MU). The CT/VT in the substation send the sampled value (analog voltage or current) to the Merging Unit (MU) over hard wire. The merging unit sends the time-synchronized 61850-9-2 sampled values to the IEDs in the substation in GOOSE message format. The GPS Master Clock can send 1PPS or IRIG-B format to MU through serial port, or IEEE 1588 protocol via network. Process bus communication using 61850 simplifies connectivity within the substation and removes the requirement for multiple serial connections and removes the slow serial bus architectures that are typically used. This also ensures increased

flexibility and increased speed with the use of multicast messaging between multiple devices.

Intra-Substation protection Requirement	Attribute
One way maximum delay	5 ms
Asymetric delay Required	No
Maximum jitter	Not critical
Topology	Point to point, point to Multi-point
Bandwidth	64 Kbps
Availability	99.9999
precise timing required	Yes
Recovery time on Node failure	less than 50ms - hitless
performance management	Yes, Mandatory
Redundancy	Yes - No
Packet loss	0.1%

Table 6: Intra-Substation Protection requirements

4.2.1.7. Wide Area Monitoring and Control Systems

The application of synchrophasor measurement data from Phasor Measurement Units (PMU) to Wide Area Monitoring and Control Systems promises to provide important new capabilities for improving system stability. Access to PMU data enables more timely situational awareness over larger portions of the grid than what has been possible historically with normal SCADA data. Handling the volume and real-time nature of synchrophasor data presents unique challenges for existing application architectures. Wide Area management System (WAMS) makes it possible for the condition of the bulk power system to be observed and understood in real-time so that protective, preventative, or corrective action can be taken. Because of the very high sampling rate of measurements and the strict requirement for time synchronization of the samples, WAMS has stringent

telecommunication requirements in an IP network that are captured in the following table:

WAMS Requirement	Attribute
One way maximum delay	50 ms
Asymetric delay Required	No
Maximum jitter	Not critical
Topology	Point to point, point to Multi-point, Multi-point to Multi-point
Bandwidth	100 Kbps
Availability	99.9999
precise timing required	Yes
Recovery time on Node failure	less than 50ms - hitless
performance management	Yes, Mandatory
Redundancy	Yes
Packet loss	1%

Table 7: WAMS Special Communication Requirements

4.2.2. Distribution use case

4.2.2.1. Fault Location Isolation and Service Restoration (FLISR)

As the name implies, Fault Location, Isolation, and Service Restoration (FLISR) refers to the ability to automatically locate the fault, isolate the fault, and restore service in the distribution network. It is a self-healing feature whose purpose is to minimize the impact of faults by serving portions of the loads on the affected circuit by switching to other circuits. It reduces the number of customers that experience a sustained power outage by reconfiguring

distribution circuits. This will likely be the first wide spread application of distributed intelligence in the grid. Secondary substations can be connected to multiple primary substations. Normally, static power switch statuses (open/closed) in the network dictate the power flow to secondary substations. Reconfiguring the network in the event of a fault is typically done manually on site to operate switchgear to energize/de-energize alternate paths. Automating the operation of substation switchgear allows the utility to have a more dynamic network where the flow of power can be altered under fault conditions but also during times of peak load. It allows the utility to shift peak loads around the network. Or, to be more precise, alters the configuration of the network to move loads between different primary substations. The FLISR capability can be enabled in two modes:

- Managed centrally from DMS, or
- Executed locally through distributed control via intelligent switches and fault sensors.

There are 3 distinct sub-functions that are performed:

1. Fault Location Identification

This sub-function is initiated by SCADA inputs, such as lockouts, fault indications/location, and, also, by input from the Outage Management System (OMS), and in the future by inputs from fault-predicting devices. It determines the specific protective device, which has cleared the sustained fault, identifies the de-energized sections, and estimates the probable location of the actual or the expected fault. It distinguishes faults cleared by controllable protective devices from those cleared by fuses, and identifies momentary outages and inrush/cold load pick-up currents. This step is also referred to as Fault Detection Classification and Location (FDCL). This step helps to expedite the restoration of faulted sections through fast fault location identification and improved diagnostic information available for crew dispatch. Also provides visualization of fault information to design and implement a switching plan to isolate the fault.

2. Fault Type Determination

I. Indicates faults cleared by controllable protective devices by distinguishing between:

a. Faults cleared by fuses

b. Momentary outages

c. Inrush/cold load current

II. Determines the faulted sections based on SCADA fault indications and protection lockout signals

III. Increases the accuracy of the fault location estimation based on SCADA fault current measurements and real-time fault analysis

3. Fault Isolation and Service Restoration

Once the location and type of the fault has been pinpointed the systems will attempt to isolate the fault and restore the non-faulted section of the network. This can have three modes of operation:

I. Closed-loop mode : This is initiated by the Fault location sub-function. It generates a switching order (i.e., sequence of switching) for the remotely controlled switching devices to isolate the faulted section, and restore service to the non-faulted sections. The switching order is automatically executed via SCADA.

II. Advisory mode : This is initiated by the Fault location sub-function. It generates a switching order for remotely and manually controlled switching devices to isolate the faulted section, and restore service to the non-faulted sections. The switching order is presented to operator for approval and execution

III. Study mode : the operator initiates this function. It analyzes a saved case modified by the operator, and generates a switching order under the operating conditions specified by the operator.

With the increasing volume of data that are collected through fault sensors, utilities will use Big Data query and analysis tools to study outage information to anticipate and prevent outages by detecting failure patterns and their correlation with asset age, type, load profiles, time of day, weather conditions, and other conditions to discover conditions that lead to faults and take the necessary preventive and corrective measures.

FLISR Requirement	Attribute
One way maximum delay	80 ms
Asymetric delay Required	No
Maximum jitter	40 ms
Topology	Point to point, point to Multi-point, Multi-point to Multi-point
Bandwidth	64 Kbps
Availability	99.9999
precise timing required	Yes
Recovery time on Node failure	Depends on customer impact
performance management	Yes, Mandatory
Redundancy	Yes
Packet loss	0.1%

Table 8: FLISR Communication Requirements

4.2.3. Generation use case

4.2.3.1. Frequency Control / Automatic Generation Control (AGC)

The system frequency should be maintained within a very narrow band. Deviations from the acceptable frequency range are detected and forwarded to the Load Frequency Control (LFC) system so that required up or down generation increase / decrease pulses can be sent to the power plants for frequency regulation. The trend in system frequency is a measure of mismatch between demand and generation, and is a necessary parameter for load control in interconnected systems.

Automatic generation control (AGC) is a system for adjusting the power output of generators at different power plants, in response to

changes in the load. Since a power grid requires that generation and load closely balance moment by moment, frequent adjustments to the output of generators are necessary. The balance can be judged by measuring the system frequency; if it is increasing, more power is being generated than used, and all machines in the system are accelerating. If the system frequency is decreasing, more demand is on the system than the instantaneous generation can provide, and all generators are slowing down.

Where the grid has tie lines to adjacent control areas, automatic generation control helps maintain the power interchanges over the tie lines at the scheduled levels. The AGC takes into account various parameters including the most economical units to adjust, the coordination of thermal, hydroelectric, and other generation types, and even constraints related to the stability of the system and capacity of interconnections to other power grids.

For the purpose of AGC we use static frequency measurements and averaging methods are used to get a more precise measure of system frequency in steady-state conditions.

During disturbances, more real-time dynamic measurements of system frequency are taken using PMUs, especially when different areas of the system exhibit different frequencies. But that is outside the scope of this use case.

FCAG Requirement	Attribute
One way maximum delay	500 ms
Asymetric delay Required	No
Maximum jitter	Not critical
Topology	Point to point
Bandwidth	20 Kbps
Availability	99.999
precise timing required	Yes
Recovery time on Node failure	N/A
performance management	Yes, Mandatory
Redundancy	Yes
Packet loss	1%

Table 9: FCAG Communication Requirements

4.3. Specific Network topologies of Smart Grid Applications

Utilities often have very large private telecommunications networks. It covers an entire territory / country. The main purpose of the network, until now, has been to support transmission network monitoring, control, and automation, remote control of generation sites, and providing FCAPS services from centralized network operation centers.

Going forward, one network will support operation and maintenance of electrical networks (generation, transmission, and distribution), voice and data services for ten of thousands of employees and for exchange with neighboring interconnections, and administrative services. To meet those requirements, utility may deploy several physical networks leveraging different technologies across the country: an optical network and a microwave network for instance. Each protection and automatism system between two points has two telecommunication circuits, one on each network. Path diversity between two substations is key. Regardless of the event type

(hurricane, ice storm, etc.), one path shall stay available so the SPS can still operate.

In the optical network, signals are transmitted over more than tens of thousands of circuits using fiber optic links, microwave and telephone cables. This network is the nervous system of the utility's power transmission operations. The optical network represents ten of thousands of km of cable deployed along the power lines.

Due to vast distances between transmission substations (as far as 280km apart), the fiber signal is amplified to reach a distance of 280 km without attenuation.

4.3.1. Precision Time Protocol

Some utilities do not use GPS clocks in generation substations. One of the main reasons is that some of the generation plants are 30 to 50 meters deep under ground and the GPS signal can be weak and unreliable. Instead, atomic clocks are used. Clocks are synchronized amongst each other. Rubidium clocks provide clock and 1ms timestamps for IRIG-B. Some companies plan to transition to the Precision Time Protocol (IEEE 1588), distributing the synchronization signal over the IP/MPLS network.

The Precision Time Protocol (PTP) is defined in IEEE standard 1588. PTP is applicable to distributed systems consisting of one or more nodes, communicating over a network. Nodes are modeled as containing a real-time clock that may be used by applications within the node for various purposes such as generating time-stamps for data or ordering events managed by the node. The protocol provides a mechanism for synchronizing the clocks of participating nodes to a high degree of accuracy and precision.

PTP operates based on the following assumptions :

It is assumed that the network eliminates cyclic forwarding of PTP messages within each communication path (e.g., by using a spanning tree protocol). PTP eliminates cyclic forwarding of PTP messages between communication paths.

PTP is tolerant of an occasional missed message, duplicated message, or message that arrived out of order. However, PTP assumes that such impairments are relatively rare.

PTP was designed assuming a multicast communication model. PTP also supports a unicast communication model as long as the behavior of the protocol is preserved.

Like all message-based time transfer protocols, PTP time accuracy is degraded by asymmetry in the paths taken by event messages. Asymmetry is not detectable by PTP, however, if known, PTP corrects for asymmetry.

A time-stamp event is generated at the time of transmission and reception of any event message. The time-stamp event occurs when the message's timestamp point crosses the boundary between the node and the network.

5. IANA Considerations

This memo includes no request to IANA.

6. Security Considerations

6.1. Current Practices and Their Limitations

Grid monitoring and control devices are already targets for cyber attacks and legacy telecommunication protocols have many intrinsic network related vulnerabilities. DNP3, Modbus, PROFIBUS/PROFINET, and other protocols are designed around a common paradigm of request and respond. Each protocol is designed for a master device such as an HMI system to send commands to subordinate slave devices to retrieve data (reading inputs) or control (writing to outputs). Because many of these protocols lack authentication, encryption, or other basic security measures, they are prone to network-based attacks, allowing a malicious actor or attacker to utilize the request-and-respond system as a mechanism for command-and-control like functionality. Specific security concerns common to most industrial control, including utility telecommunication protocols include the following:

- Network or transport errors (e.g. malformed packets or excessive latency) can cause protocol failure.
- Protocol commands may be available that are capable of forcing slave devices into inoperable states, including powering-off devices, forcing them into a listen-only state, disabling alarming.
- Protocol commands may be available that are capable of restarting communications and otherwise interrupting processes.
- Protocol commands may be available that are capable of clearing, erasing, or resetting diagnostic information such as counters and diagnostic registers.

- Protocol commands may be available that are capable of requesting sensitive information about the controllers, their configurations, or other need-to-know information.

- Most protocols are application layer protocols transported over TCP; therefore it is easy to transport commands over non-standard ports or inject commands into authorized traffic flows.

- Protocol commands may be available that are capable of broadcasting messages to many devices at once (i.e. a potential DoS).

- Protocol commands may be available to query the device network to obtain defined points and their values (i.e. a configuration scan).

- Protocol commands may be available that will list all available function codes (i.e. a function scan).

- Bump in the wire (BITW) solutions : A hardware device is added to provide IPSec services between two routers that are not capable of IPSec functions. This special IPsec device will intercept then intercept outgoing datagrams, add IPSec protection to them, and strip it off incoming datagrams. BITW can all IPSec to legacy hosts and can retrofit non-IPSec routers to provide security benefits. The disadvantages are complexity and cost.

These inherent vulnerabilities, along with increasing connectivity between IT an OT networks, make network-based attacks very feasible. Simple injection of malicious protocol commands provides control over the target process. Altering legitimate protocol traffic can also alter information about a process and disrupt the legitimate controls that are in place over that process. A man- in-the-middle attack could provide both control over a process and misrepresentation of data back to operator consoles.

6.2. Security Trends in Utility Networks

Although advanced telecommunication networks can assist in transforming the energy industry, playing a critical role in maintaining high levels of reliability, performance, and manageability, they also introduce the need for an integrated security infrastructure. Many of the technologies being deployed to support smart grid projects such as smart meters and sensors can increase the vulnerability of the grid to attack. Top security concerns for utilities migrating to an intelligent smart grid telecommunications platform center on the following trends:

- Integration of distributed energy resources
- Proliferation of digital devices to enable management, automation, protection, and control
- Regulatory mandates to comply with standards for critical infrastructure protection
- Migration to new systems for outage management, distribution automation, condition-based maintenance, load forecasting, and smart metering
- Demand for new levels of customer service and energy management

This development of a diverse set of networks to support the integration of microgrids, open-access energy competition, and the use of network-controlled devices is driving the need for a converged security infrastructure for all participants in the smart grid, including utilities, energy service providers, large commercial and industrial, as well as residential customers. Securing the assets of electric power delivery systems, from the control center to the substation, to the feeders and down to customer meters, requires an end-to-end security infrastructure that protects the myriad of telecommunication assets used to operate, monitor, and control power flow and measurement. Cyber security refers to all the security issues in automation and telecommunications that affect any functions related to the operation of the electric power systems. Specifically, it involves the concepts of:

- Integrity : data cannot be altered undetectably
- Authenticity : the telecommunication parties involved must be validated as genuine
- Authorization : only requests and commands from the authorized users can be accepted by the system
- Confidentiality : data must not be accessible to any unauthenticated users

When designing and deploying new smart grid devices and telecommunication systems, it’s imperative to understand the various impacts of these new components under a variety of attack situations on the power grid. Consequences of a cyber attack on the grid telecommunications network can be catastrophic. This is why security for smart grid is not just an ad hoc feature or product, it’s a complete framework integrating both physical and Cyber security requirements and covering the entire smart grid networks from

generation to distribution. Security has therefore become one of the main foundations of the utility telecom network architecture and must be considered at every layer with a defense-in-depth approach. Migrating to IP based protocols is key to address these challenges for two reasons:

1. IP enables a rich set of features and capabilities to enhance the security posture

2. IP is based on open standards, which allows interoperability between different vendors and products, driving down the costs associated with implementing security solutions in OT networks.

Securing OT telecommunication over packet-switched IP networks follow the same principles that are foundational for securing the IT infrastructure, i.e., consideration must be given to enforcing electronic access control for both person-to-machine and machine-to-machine communications, and providing the appropriate levels of data privacy, device and platform integrity, and threat detection and mitigation.

7. Acknowledgements

Faramarz Maghsoodlou, Ph. D. IoT Connected Industries and Energy Practice Cisco

Pascal Thubert, CTAO Cisco

8. References

8.1. Normative References

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

[min_ref] authSurName, authInitials., "Minimal Reference", 2006.

8.2. Informative References

[I-D.finn-detnet-problem-statement]
Finn, N. and P. Thubert, "Deterministic Networking Problem Statement", draft-finn-detnet-problem-statement-01 (work in progress), October 2014.

Authors' Addresses

Patrick Wetterwald
Cisco Systems
45 Allees des Ormes
Mougins 06250
FRANCE

Phone: +33 4 97 23 26 36
Email: pwetterw@cisco.com

Jean Raymond
Hydro-Quebec
1500 University
Montreal H3A3S7
Canada

Phone: +1 514 840 3000
Email: raymond.jean@hydro.qc.ca