

6TiSCH
Internet-Draft
Intended status: Informational
Expires: December 13, 2015

P. Thubert, Ed.
Cisco
June 11, 2015

6TiSCH requirements for DetNet
[draft-thubert-6tisch-4detnet-01](#)

Abstract

This document builds on the 6TiSCH architecture that defines, among others, mechanisms to establish and maintain deterministic routing and scheduling in a centralized fashion. The document details dependencies on DetNet and PCE controller to express topologies and capabilities, as well as abstract state that the controller must be able to program into the network devices to enable deterministic forwarding operations.

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 December 13, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

Internet-Draft

6tisch-4detnet

June 2015

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

Table of Contents

1.	Introduction	2
2.	Terminology	4
3.	6TiSCH Overview	4
3.1.	TSCH and 6top	7
3.2.	SlotFrames and Priorities	7
3.3.	Schedule Management by a PCE	9
3.4.	Track Forwarding	10
3.4.1.	Transport Mode	11
3.4.2.	Tunnel Mode	12
3.4.3.	Tunnel Metadata	13
4.	Operations of Interest for DetNet and PCE	14
4.1.	Packet Marking and Handling	15
4.1.1.	Tagging Packets for Flow Identification	15
4.1.2.	Replication, Retries and Elimination	15
4.1.3.	Differentiated Services Per-Hop-Behavior	16
4.2.	Topology and capabilities	16
5.	IANA Considerations	17
6.	Security Considerations	17
7.	Acknowledgments	17
8.	References	18
8.1.	Normative References	18
8.2.	Informative References	18
8.3.	Other Informative References	20
	Author's Address	21

[1.](#) Introduction

The emergence of wireless technology has enabled a variety of new devices to get interconnected, at a very low marginal cost per device, at any distance ranging from Near Field to interplanetary, and in circumstances where wiring may not be practical, for instance

on fast-moving or 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 Medium access Control (MAC) of IEEE802.15.4 [[IEEE802154](#)] has evolved with the new TimeSlotted Channel Hopping (TSCH) [[I-D.ietf-6tisch-tsch](#)] mode for deterministic industrial-type applications. TSCH was introduced with the IEEE802.15.4e [[IEEE802154e](#)] amendment and will be wrapped up in the next revision of the IEEE802.15.4 standard. For all practical purpose, this document is expected to be insensitive to the future versions of the IEEE802.15.4 standard, which is thus referenced undated.

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 multi-path fading and co-channel interferences (for example by Wi-Fi emitters).

The 6TiSCH Architecture [[I-D.ietf-6tisch-architecture](#)] defines a remote monitoring and scheduling management of a TSCH network by a Path Computation Element (PCE), which cooperates with an abstract Network Management Entity (NME) to manage timeSlots and device resources in a manner that minimizes the interaction with and the load placed on the constrained devices.

This Architecture applies the concepts of Deterministic Networking on a TSCH network to enable the switching of timeSlots in a G-MPLS manner. This document details the dependencies that 6TiSCH has on PCE [[PCE](#)] and DetNet [[I-D.finn-detnet-architecture](#)] to provide the necessary capabilities that may be specific to such networks. In turn, DetNet is expected to integrate and maintain consistency with the work that has taken place and is continuing at IEEE802.1TSN and AVnu.

2. Terminology

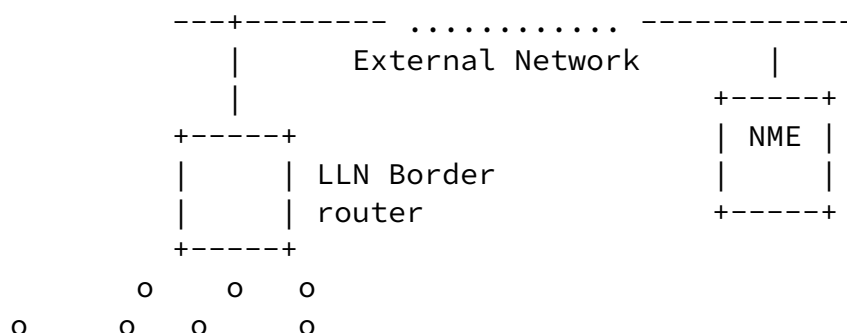
Readers are expected to be familiar with all the terms and concepts that are discussed in "Multi-link Subnet Support in IPv6" [[I-D.ietf-ipv6-multilink-subnets](#)].

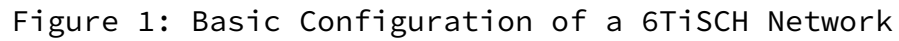
The draft uses terminology defined or referenced in [[I-D.ietf-6tisch-terminology](#)] and [[I-D.ietf-roll-rpl-industrial-applicability](#)].

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

3. 6TiSCH Overview

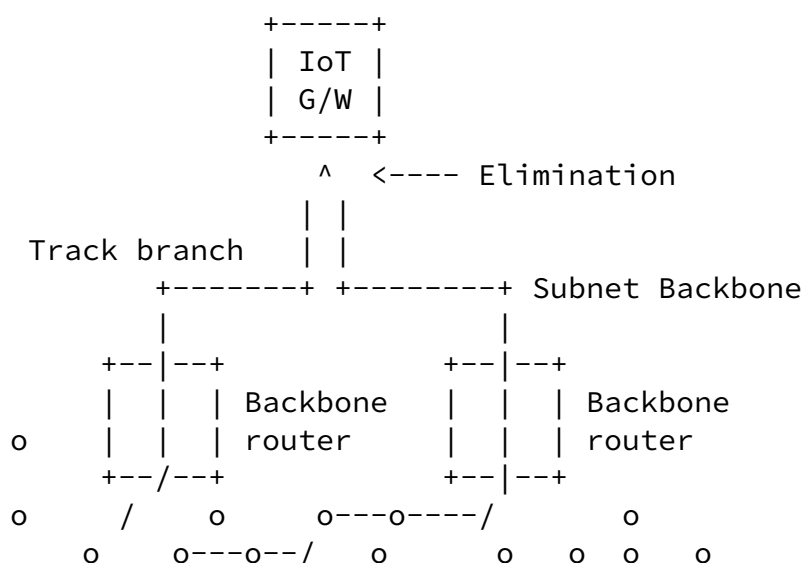
The scope of the present work is a subnet that, in its basic configuration, is made of a TSCH [[I-D.ietf-6tisch-tsch](#)] MAC Low Power Lossy Network (LLN).





If the Backbone is Deterministic, then the Backbone Router ensures that the end-to-end deterministic behavior is maintained between the LLN and the backbone. This SHOULD be done in conformance to the DetNet Architecture [[I-D.finn-detnet-architecture](#)] which studies Layer-3 aspects of Deterministic Networks, and covers networks that span multiple Layer-2 domains. One particular requirement is that the PCE MUST be able to compute a deterministic path and to end across the TSCH network and an IEEE802.1 TSN Ethernet backbone, and DetNet MUST enable end-to-end deterministic forwarding.

6TiSCH defines the concept of a Track, which is a complex form of a uni-directional Circuit ([[I-D.ietf-6tisch-terminology](#)]). As opposed to a simple circuit that is a sequence of nodes and links, a Track is shaped as a directed acyclic graph towards a destination to support multi-path forwarding and route around failures. A Track may also branch off and rejoin, for the purpose of the so-called Packet Replication and Elimination (PRE), over non congruent branches. PRE may be used to complement layer-2 Automatic Repeat reQuest (ARQ) to meet industrial expectations in Packet Delivery Ratio (PDR), in particular when the Track extends beyond the 6TiSCH network.



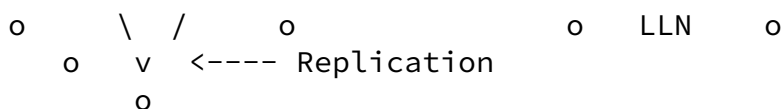


Figure 3: End-to-End deterministic Track

In the example above, a Track is laid out from a field device in a 6TiSCH network to an IoT gateway that is located on a IEEE802.1 TSN backbone.

The Replication function in the field device sends a copy of each packet over two different branches, and the PCE schedules each hop of both branches so that the two copies arrive in due time at the gateway. In case of a loss on one branch, hopefully the other copy of the packet still makes it in due time. If two copies make it to the IoT gateway, the Elimination function in the gateway ignores the extra packet and presents only one copy to upper layers.

At each 6TiSCH hop along the Track, the PCE may schedule more than one timeSlot for a packet, so as to support Layer-2 retries (ARQ). It is also possible that the field device only uses the second branch if sending over the first branch fails.

In current deployments, a TSCH Track does not necessarily support PRE but is systematically multi-path. This means that a Track is scheduled so as to ensure that each hop has at least two forwarding solutions, and the forwarding decision is to try the preferred one and use the other in case of Layer-2 transmission failure as detected by ARQ.

3.1. TSCH and 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 [\[I-D.wang-6tisch-6top-sublayer\]](#).

The 6top data model and management interfaces are further discussed

in [[I-D.ietf-6tisch-6top-interface](#)] and [[I-D.ietf-6tisch-coap](#)].

The architecture defines "soft" cells and "hard" cells. "Hard" cells are owned and managed by an separate scheduling entity (e.g. a PCE) that specifies the slotOffset/channelOffset of the cells to be added/moved/deleted, in which case 6top can only act as instructed, and may not move hard cells in the TSCH schedule on its own.

[3.2.](#) SlotFrames and Priorities

A slotFrame is the base object that the PCE needs to manipulate to program a schedule into an LLN node. Elaboration on that concept can be found in section "SlotFrames and Priorities" of [[I-D.ietf-6tisch-architecture](#)]

IEEE802.15.4 TSCH avoids contention on the medium by formatting time and frequencies in cells of transmission of equal duration. 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 802.15.4 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.

The frequency used by a cell in the matrix rotates in a pseudo-random fashion, from an initial position at an epoch time, as the matrix iterates over and over.

A CDU matrix is computed by the PCE, but unallocated timeSlots may be used opportunistically by the nodes for classical best effort IP traffic. The PCE has precedence in the allocation in case of a conflict.

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 PCE MUST compute the CDU matrices and shared that knowledge with all the nodes. The matrices are 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.

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 PCE MUST compute the partitioning of CDU matrices into chunks and shared that knowledge with all the nodes in a 6TiSCH network.

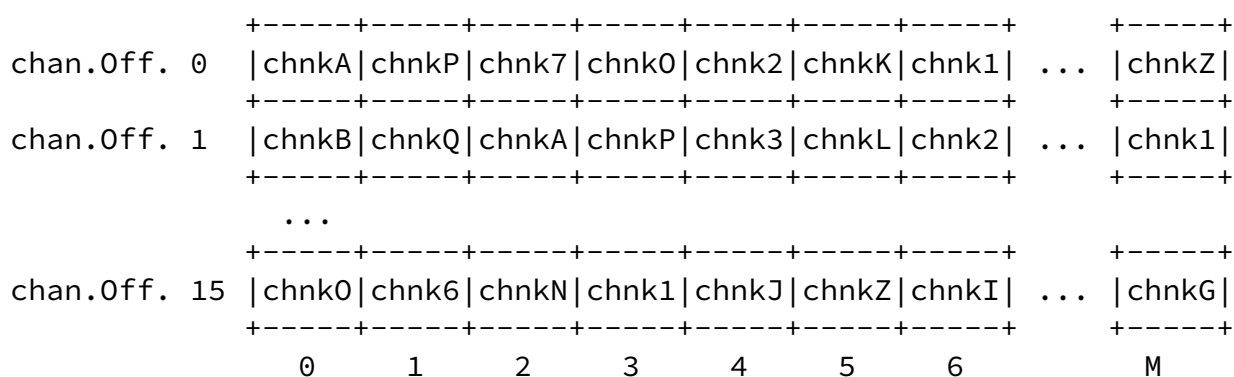


Figure 4: CDU matrix Partitioning in Chunks

The appropriation of a chunk can be requested explicitly by the PCE to any node. After a successful appropriation, the PCE owns the cells in that chunk, and may use them as hard cells to set up Tracks.

[3.3.](#) Schedule Management by a PCE

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.

Section "Schedule Management Mechanisms" of the 6TiSCH architecture describes 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. The Track operation for DetNet corresponds to a remote monitoring and scheduling management by a PCE.

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.

[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] also 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 PCE commands are expected to be issued directly as CoAP requests or to be mapped back and forth into CoAP by a gateway function at the edge of the 6TiSCH network. For instance, it is possible that a mapping entity on the backbone transforms a non-CoAP protocol such as PCEP into the RESTful interfaces that the 6TiSCH devices support.

This architecture will be refined to comply with DetNet
[[I-D.finn-detnet-architecture](#)] when the work is formalized.

[3.4.](#) Track Forwarding

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 the routing computation of a Track by a PCE. The 6TiSCH architecture supports three different forwarding model, G-MPLS Track Forwarding (TF), 6LoWPAN Fragment Forwarding (FF) and IPv6 Forwarding (6F) which is the classical IP operation. The DetNet case relates to the Track Forwarding operation under the control of a PCE.

A Track is a unidirectional path between a source and a destination. In a Track cell, the normal operation of IEEE802.15.4 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 IEEE802.15.4, 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

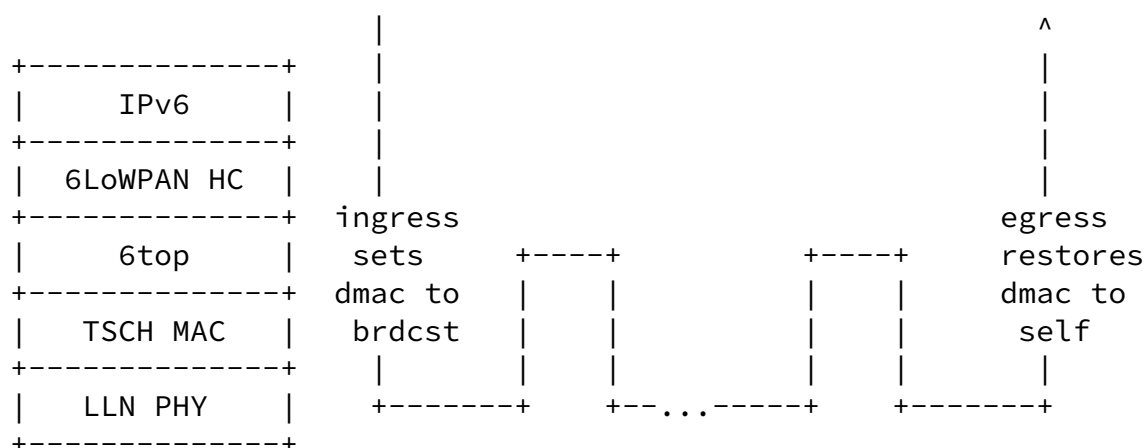
makes sense for the receiver to recognize a frame that should be re-tracked, and to place it back on the appropriate bundle if possible. A frame should be re-tracked if the Per-Hop-Behavior group indicated in the Differentiated Services Field in the IPv6 header is set to Deterministic Forwarding, as discussed in [Section 4.1](#). 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.

[3.4.1.](#) 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.



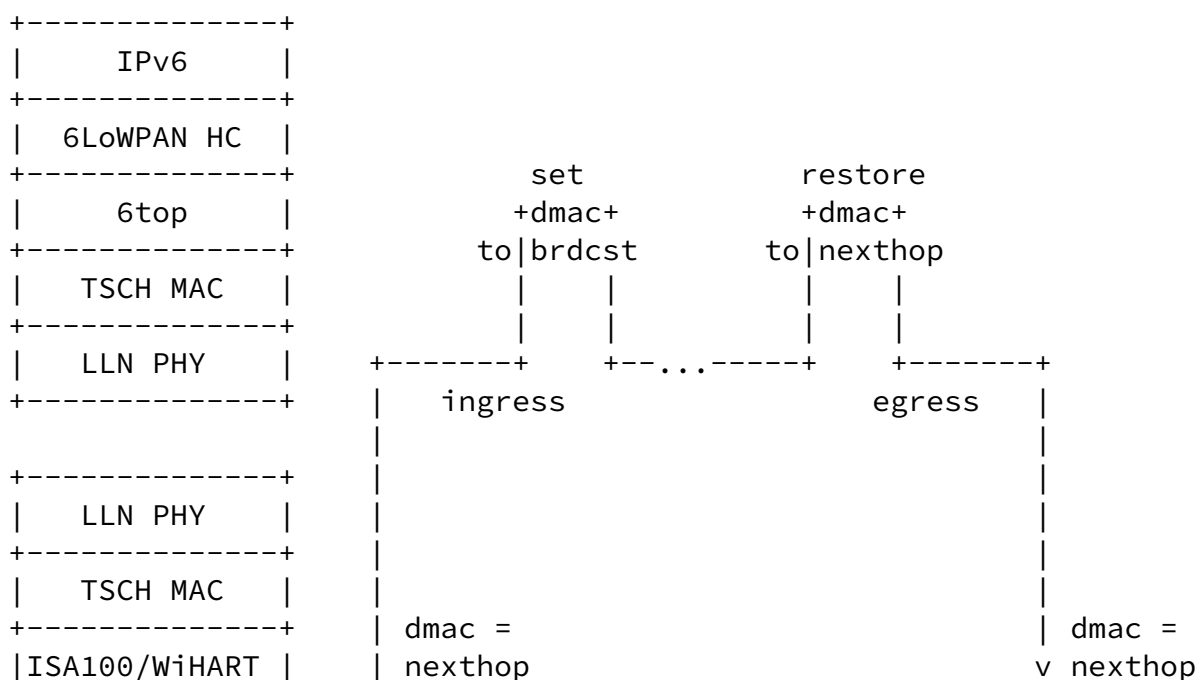
Track Forwarding, Transport Mode

[3.4.2.](#) Tunnel Mode

In tunnel mode, the frames originate from an arbitrary protocol over

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

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



+-----+

Figure 5: Track Forwarding, Tunnel Mode

In that case, the flow information that identifies the Track at the ingress 6TiSCH router is derived from the RX-cell. The dmac is set to this node but the flow information indicates that the frame must be 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.

[3.4.3.](#) 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.

[4.](#) Operations of Interest for DetNet and PCE

In a classical system, the 6TiSCH device does not place the request for bandwidth between self and another device in the network. Rather, an Operation Control System invoked through an Human/Machine Interface (HMI) indicates the Traffic Specification, in particular in

terms of latency and reliability, and the end nodes. With this, the PCE must compute a Track between the end nodes and provision the network with per-flow state that describes the per-hop operation for a given packet, the corresponding timeSlots, and the flow identification that enables to recognize when a certain packet belongs to a certain Track, sort out duplicates, etc...

For a static configuration that serves a certain purpose for a long period of time, it is expected that a node will be provisioned in one shot with a full schedule, which incorporates the aggregation of its behavior for multiple Tracks. 6TiSCH expects that the programming of the schedule will be done over COAP as discussed in 6TiSCH Resource Management and Interaction using CoAP [[I-D.ietf-6tisch-coap](#)].

But an Hybrid mode may be required as well whereby a single Track is added, modified, or removed, for instance if it appears that a Track does not perform as expected for, say, PDR. For that case, the expectation is that a protocol that flows along a Track (to be), in a fashion similar to classical Traffic Engineering (TE) [[CCAMP](#)], may be used to update the state in the devices. 6TiSCH provides means for a device to negotiate a timeSlot with a neighbor, but in general that flow was not designed and no protocol was selected and it is expected that DetNet will determine the appropriate end-to-end protocols to be used in that case.

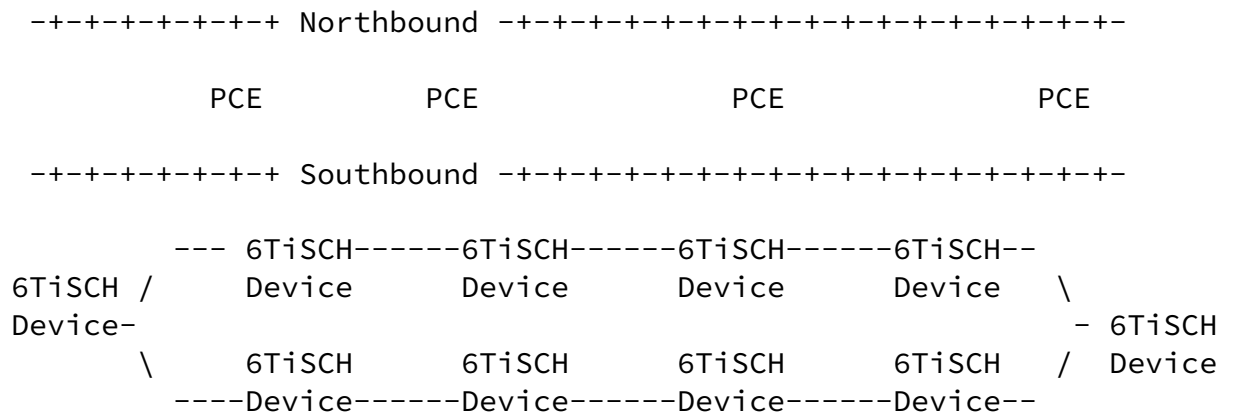


Figure 6

[4.1. Packet Marking and Handling](#)

Section "Packet Marking and Handling" of [\[I-D.ietf-6tisch-architecture\]](#) describes the packet tagging and marking that is expected in 6TiSCH networks.

[4.1.1. Tagging Packets for Flow Identification](#)

For packets that are routed by a PCE along a Track, the tuple formed by the IPv6 source address and a local RPLInstanceID is tagged in the packets to identify uniquely the Track and associated transmit bundle of timeSlots.

It results that the tagging that is used for a DetNet flow outside the 6TiSCH LLN MUST be swapped into 6TiSCH formats and back as the packet enters and then leaves the 6TiSCH network.

Note: The method and format used for encoding the RPLInstanceID at 6lo is generalized to all 6TiSCH topological Instances, which includes Tracks.

[4.1.2. Replication, Retries and Elimination](#)

6TiSCH expects elimination and replication of packets along a complex Track, but has no position about how the sequence numbers would be tagged in the packet.

As it goes, 6TiSCH expects that timeSlots corresponding to copies of a same packet along a Track are correlated by configuration, and does not need to process the sequence numbers.

The semantics of the configuration MUST enable correlated timeSlots to be grouped for transmit (and respectively receive) with a 'OR' relations, and then a 'AND' relation MUST be configurable between groups. The semantics is that if the transmit (and respectively receive) operation succeeded in one timeSlot in a 'OR' group, then all the other timeSlots in the group are ignored. Now, if there are at least two groups, the 'AND' relation between the groups indicates that one operation must succeed in each of the groups.

On the transmit side, timeSlots provisioned for retries along a same branch of a Track are placed a same 'OR' group. The 'OR' relation indicates that if a transmission is acknowledged, then further transmissions SHOULD NOT be attempted for timeSlots in that group. There are as many 'OR' groups as there are branches of the Track departing from this node. Different 'OR' groups are programmed for the purpose of replication, each group corresponding to one branch of the Track. The 'AND' relation between the groups indicates that transmission over any of branches MUST be attempted regardless of whether a transmission succeeded in another branch. It is also possible to place cells to different next-hop routers in a same 'OR' group. This allows to route along multi-path tracks, trying one next-hop and then another only if sending to the first fails.

On the receive side, all timeSlots are programmed in a same 'OR' group. Retries of a same copy as well as converging branches for elimination are converged, meaning that the first successful reception is enough and that all the other timeSlots can be ignored.

[4.1.3.](#) Differentiated Services Per-Hop-Behavior

Additionally, an IP packet that is sent along a Track uses the Differentiated Services Per-Hop-Behavior Group called Deterministic Forwarding, as described in [\[I-D.svshah-tsvwg-deterministic-forwarding\]](#).

[4.2.](#) Topology and capabilities

6TiSCH nodes are usually IoT devices, characterized by very limited amount of memory, just enough buffers to store one or a few IPv6 packets, and limited bandwidth between peers. It results that a node will maintain only a small number of peering information, and will not be able to store many packets waiting to be forwarded. Peers can be identified through MAC or IPv6 addresses, but a Cryptographically

Generated Address [[RFC3972](#)] (CGA) may also be used.

Neighbors can be discovered over the radio using mechanism such as beacons, but, though the neighbor information is available in the 6TiSCH interface data model, 6TiSCH does not describe a protocol to pro-actively push the neighborhood information to a PCE. This protocol should be described and should operate over CoAP. The protocol should be able to carry multiple metrics, in particular the same metrics as used for RPL operations [[RFC6551](#)]

The energy that the device consumes in sleep, transmit and receive modes can be evaluated and reported. So can the amount of energy that is stored in the device and the power that it can be scavenged from the environment. The PCE SHOULD be able to compute Tracks that will implement policies on how the energy is consumed, for instance balance between nodes, ensure that the spent energy does not exceeded the scavenged energy over a period of time, etc...

[5.](#) IANA Considerations

This specification does not require IANA action.

[6.](#) Security Considerations

On top of the classical protection of control signaling that can be expected to support DetNet, it must be noted that 6TiSCH networks operate on limited resources that can be depleted rapidly if an attacker manages to operate a DoS attack on the system, for instance by placing a rogue device in the network, or by obtaining management control and to setup extra paths.

[7.](#) Acknowledgments

This specification derives from the 6TiSCH architecture, which 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: Kris Pister, Thomas Watteyne, Xavier Vilajosana, Qin Wang, Tom Phinney, Robert Assimiti, Michael Richardson, Zhuo Chen, Malisa Vucinic, Alfredo Grieco, Martin Turon, Dominique Barthel, Elvis Vogli, Guillaume Gaillard, Herman Storey,

Maria Rita Palattella, Nicola Accettura, Patrick Wetterwald, Pouria Zand, Raghuram Sudhaakar, and Shitanshu Shah for their participation and various contributions.

Thubert

Expires December 13, 2015

[Page 17]

Internet-Draft

6tisch-4detnet

June 2015

[8.](#) References

[8.1.](#) Normative References

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

Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top) Interface", [draft-ietf-6tisch-6top-interface-03](#) (work in progress), March 2015.

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

Thubert, P., "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4", [draft-ietf-6tisch-architecture-08](#) (work in progress), May 2015.

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

Sudhaakar, R. and P. Zand, "6TiSCH Resource Management and Interaction using CoAP", [draft-ietf-6tisch-coap-03](#) (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-04](#) (work in progress), March 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-06](#) (work in progress), March 2015.

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

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

[8.2.](#) Informative References

[I-D.finn-detnet-architecture]

Finn, N., Thubert, P., and M. Teener, "Deterministic Networking Architecture", [draft-finn-detnet-architecture-01](#) (work in progress), March 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.

Thubert	Expires December 13, 2015	[Page 18]
---------	---------------------------	-----------

Internet-Draft	6tisch-4detnet	June 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.

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

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

[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.

- [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.
- [RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", [RFC 3972](#), March 2005.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", [RFC 4291](#), February 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.

Thubert

Expires December 13, 2015

[Page 19]

Internet-Draft

6tisch-4detnet

June 2015

- [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.
- [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.
- [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.

[8.3.](#) Other Informative References

- [ACE] IETF, "Authentication and Authorization for Constrained Environments", <<https://datatracker.ietf.org/doc/charter-ietf-ace/>>.
- [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>>.
- [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".

Thubert

Expires December 13, 2015

[Page 20]

Internet-Draft

6tisch-4detnet

June 2015

- [IEEE802154e] IEEE standard for Information Technology, "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 as amended by 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.

Author's Address

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

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