6TiSCH P. Thubert, Ed.
Internet-Draft Cisco
Intended status: Informational June 10, 2016
Expires: December 12, 2016

An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4
draft-ietf-6tisch-architecture-10

Abstract

This document describes a network architecture that provides low-latency, low-jitter and high-reliability packet delivery. It combines a high speed powered backbone and subnetworks using IEEE 802.15.4 time-slotted channel hopping (TSCH) to meet the requirements of LowPower wireless deterministic applications.

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 12, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction

Wireless Networks enable a wide variety of devices of any size to get interconnected, often at a very low marginal cost per device, at any distance ranging from Near Field to interplanetary, and in circumstances where wiring may be impractical, for instance on fast-moving or rotating devices.

In the other hand, Deterministic Networks 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. Applications that need such networks are presented in [I-D.ietf-detnet-use-cases]. They include Professional Media and Operation Technology (OT) Industrial Automation Control Systems (IACS).

The Medium access Control (MAC) of IEEE802.15.4 [IEEE802154] has evolved with the IEEE802.15.4e Timeslotted Channel Hopping (TSCH) [RFC7554] mode to provide deterministic properties on wireless networks. TSCH was initially introduced with the IEEE802.15.4e amendment [IEEE802154e] of the IEEE802.15.4 standard and constituted a part of the standard from that day. For all practical purpose, this document is expected to be insensitive to the revisions of the IEEE802.15.4 standard, which is thus referenced undated.

Proven Deterministic Networking standards for use in Process Control, including ISA100.11a [ISA100.11a] and WirelessHART [WirelessHART], have demonstrated the capabilities of the IEEE802.15.4 TSCH MAC for high reliability against interference, low-power consumption on well-known flows, and its applicability for Traffic Engineering (TE) from a central controller.

In order to enable the convergence of IT and OT in LLN environments, 6TiSCH ports the IETF suite of protocol that are defined for such environments over the TSCH MAC. 6TiSCH also provides large scaling capabilities, which, in a number of scenarios, require the addition of a high speed and reliable backbone and the use of IP version 6 (IPv6). The 6TiSCH Architecture introduces an IPv6 Multi-Link subnet model that is composed of a federating backbone and a number of IEEE802.15.4 TSCH low-power wireless networks attached and synchronized by Backbone Routers.

The architecture defines mechanisms to establish and maintain routing and scheduling in a centralized, distributed, or mixed fashion, for use in multiple OT environments. It is applicable in particular to industrial control systems, building automation that leverage

distributed routing 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.).

2. Terminology

The draft uses domain-specific terminology defined or referenced in [I-D.ietf-6tisch-terminology], [I-D.ietf-6lo-backbone-router], and [I-D.ietf-roll-rpl-industrial-applicability].

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], and Neighbor Discovery Optimization for Low-power and Lossy Networks [RFC6775] where the 6LoWPAN Router (6LR) and the 6LoWPAN Border Router (6LBR) are introduced.

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 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 and refers [RFC4080] for reservation signaling and [RFC5191] for authentication.

3. High Level Architecture

3.1. 6TiSCH Stack

The 6TiSCH architecture presents a reference stack that is implemented and interop tested by a conjunction of opensource, IETF and ETSI efforts. One goal is to help other bodies to adopt the stack as a whole, making the effort to move to an IPv6-based IOT stack easier. Now, for a particular, environment, some of the choices that are made in this architecture may not be relevant. For instance, RPL is not required for star topologies and mesh-under layer-2 routed networks, and the 6LoWPAN compression may not be

sufficient for ultra-constrained cases such as some Low Power Wide Area (LPWA) networks. In such cases, it is perfectly doable to adopt a subset of the selection that is presented hereafter and then select alternate components to complete the solution wherever needed.

The IETF proposes multiple techniques for implementing functions related to routing, transport or security. In order to control the complexity of the possible deployments and device interactions, and to limit the size of the resulting object code, the architecture limits the possible variations of the stack and recommends a number of base elements for LLN applications. In particular, UDP [RFC0768] [RFC2460] and the Constrained Application Protocol [RFC7252] (CoAP) are used as the transport / binding of choice for applications and management as opposed to TCP and HTTP.

The resulting stack is represented below:

```
+-----+-----+-----+------+-------+-----+
|     (COMI)      |(PANA)|6LoWPAN| RPL |
| CoAP  / DTLS    |      |   ND  |     |
+-----+-----+-----+------+-------+-----+
|       UDP       |          ICMP      |
+-----+-----+-----+-----+-------+------+-----+
|                 IPv6                       |
+--------------------------------------------+
|  6LoWPAN adaptation and compression (HC)   |
+--------------------------------------------+
|                     6top                   |
+--------------------------------------------+
|              IEEE802.15.4    TSCH          |
+--------------------------------------------+
```

Figure 1: 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 6LoWPAN Header Compression [RFC6282] is used to compress the IPv6 and UDP headers, whereas the 6LoWPAN Routing Header [I-D.ietf-roll-routing-dispatch] is used to compress the RPL artifacts in the IPv6 data packets, including the RPL Packet Information (RPI), the IP-in-IP encapsulation to/from the RPL root, and the Source Route Header (SRH) in non-storing mode.

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].

The Datagram Transport Layer Security (DTLS) [RFC6347] is represented as an example of a protocol that could be used to protect CoAP datagrams, but the exact stack is not determined at the time of this writing..

Similarly, the Protocol for Carrying Authentication for Network access (PANA) [RFC5191] is represented as an example of a protocol that could be leveraged to secure the join process, as a Layer-3 alternate to IEEE802.1x/EAP. Regardless, the security model ensures that, prior to a join process, packets from a untrusted device are controlled in volume and in reachability. In particular, a PANA stack should be separated from the main protocol stack to avoid attacks during the join process that is introduced in Section 3.7. An overview of the security aspects of the join process can be found in Section 6.

The 6TiSCH Operation sublayer (6top) [I-D.wang-6tisch-6top-sublayer] is a sublayer of a Logical Link Control (LLC) that provides the abstraction of an IP link over a TSCH MAC and schedules packets over TSCH cells,as further discussed in the next sections.

3.2. TSCH: A Deterministic MAC Layer

Though at a different time scale (several orders of magnitude), both IEEE802.1TSN and IEEE802.15.4TSCH standards provide Deterministic capabilities to the point that a packet that pertains to a certain flow may traverse a network from node to node following a very precise schedule, as a train that enters and then leaves intermediate stations at precise times along its path. With TSCH, time is formatted into timeslots, and individual communication cells are 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).

6TiSCH builds on the IEEE802.15.4TSCH MAC and inherits its advanced capabilities to enable them in multiple environments where they can be leveraged to improve automated operations. The 6TiSCH

Architecture also inherits the capability to perform a centralized route computation to achieve deterministic properties, though it relies on the IETF DetNet Architecture [I-D.finn-detnet-architecture], and IETF components such as the Path Computation Element (PCE) [PCE], for the protocol aspects.

On top of this inheritance, 6TiSCH adds capabilities for distributed routing and scheduling operations based on the RPL routing protocol and capabilities to negotiate schedule adjustments between peers. These distributed routing and scheduling operations simplify the deployment of TSCH networks and enable wireless solutions in a larger variety of use cases from operational technology in general. Examples of such use-cases in industrial environments 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 presented in [I-D.ietf-roll-rpl-industrial-applicability].

3.3. Scheduling TSCH

A scheduling operation attributes cells in a Time-Division-Multiplexing (TDM) / Frequency-Division Multiplexing (FDM) matrix called the Channel distribution/usage (CDU) to either individual transmissions or as multi-access shared resources (see the 6TiSCH Terminology [I-D.ietf-6tisch-terminology] for more on these terms). Scheduling effectively enables multiple communications at a same time in a same interference domain using different channels; but a node equipped with a single radio can only transmit or receive on one channel at any given point of time.

From the standpoint of a 6TiSCH node (at the MAC layer), its schedule is the collection of the times at which it must wake up for transmission, and the channels to which it should either send or listen at those times. The schedule is expressed as one or more slotframes that repeat over and over. Slotframes may collision and require a device to wake at a same time, in which case a priority indicates which slotframe is actually activated.

The 6top sublayer hides the complexity of the schedule to the upper layers. The Link that IP may utilize between the 6TiSCH node and a peer may in fact be composed of a pair of cell bundles, one to receive and one to transmit. Some of the cells may be shared, in which case the 6top sublayer must perform some arbitration.

The 6TiSCH architecture identifies four ways a schedule can be managed and CDU cells can be allocated: Static Scheduling, Neighbor-to-Neighbor Scheduling, Remote Monitoring and Schedule Management, and Hop-by-hop Scheduling.

Static Scheduling: This refers to the minimal 6TiSCH operation whereby a static schedule is configured for the whole network for use in a slotted-aloha fashion. The static schedule is distributed through the native methods in the TSCH MAC layer. This operation leverages RPL to maintain a loopless graph for routing and time distribution. It is specified in the Minimal 6TiSCH Configuration [I-D.ietf-6tisch-minimal] specification. and does not preclude other scheduling operations to co-exist on a same 6TiSCH network.

Neighbor-to-Neighbor Scheduling: This refers to the dynamic adaptation of the bandwidth of the Links that are used for IPv6 traffic between adjacent routers. Scheduling Functions such as SF0 [I-D.ietf-6tisch-6top-sf0] influence the operation of the 6top sublayer [I-D.wang-6tisch-6top-sublayer] to add and remove cells in peers schedule, using the 6top protocol [I-D.ietf-6tisch-6top-protocol] for the negotiation on the MAC resources.

Remote Monitoring and Schedule Management: This refers to the central computation of a schedule and the capability to forward a frame based on the cell of arrival. In that case, the related portion of the device schedule as well as 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. This model is the TSCH adaption of the DetNet Architecture [I-D.finn-detnet-architecture], and it enables Traffic Engineering with deterministic properties.

Hop-by-hop Scheduling: This refers to the possibility to reserves cells along a path for a particular flow using a distributed mechanism.

It is not expected that all use cases will require all those mechanisms. Static Scheduling with minimal configuration one is the only one that is expected in all implementations, since it provides a simple and solid basis for convergecast routing and time distribution.

A deeper dive in those mechanisms can be found in Section 4.4.

3.4. Routing and Forwarding Over TSCH

6TiSCH leverages the RPL routing protocol for interoperable distributed routing operations. RPL is applicable to Static Scheduling and Neighbor-to-Neighbor Scheduling. The architecture also supports a centralized routing model for Remote Monitoring and

Schedule Management. It is expected that a routing protocol that is more optimized for point-to-point routing than RPL, such as the Reactive Discovery of Point-to-Point Routes in Low-Power and Lossy Networks [RFC6997](P2P RPL), or the Ad Hoc On-demand Distance Vector Routing (AODV) [I-D.ietf-manet-aodvv2] will be selected for Hop-by-hop Scheduling.

The 6TiSCH architecture supports three different forwarding models, the classical IPv6 Forwarding, where the node selects a feasible successor at Layer-3 on a per packet basis and based on its routing table, G-MPLS Track Forwarding, which switches a frame received at a particular Timeslot into another Timeslot at Layer-2, and 6LoWPAN Fragment Forwarding, which allows to forward individual 6loWPAN fragments along the route set by the first fragment.

IPv6 Forwarding: This is the classical IP forwarding model, with a Routing Information Based (RIB) that is installed by the RPL routing protocol and used to select a feasible successor per packet. The packet is placed on an outgoing Link, that the 6top layer maps into a (Layer-3) bundle of cells, and scheduled for transmission based on QoS parameters. On top of RPL, this model also applies to any routing protocol which may be operated in the 6TiSCH network, and corresponds to all the distributed scheduling models, Static, Neighbor-to-Neighbor and Hop-by-Hop Scheduling.

G-MPLS Track Forwarding: This model corresponds to the Remote Monitoring and Schedule Management. In this model, A central controller (hosting a PCE) computes and installs the schedules in the devices per flow. The incoming (Layer-2) bundle of cells from the previous node along the path determines the outgoing (Layer-2) bundle towards the next hop for that flow as determined by the PCE. The programmed sequence for bundles is called a Track and can assume shapes that are more complex than a simple direct sequence of nodes.

6LoWPAN Fragment Forwarding: This is an hybrid model that derives from IPv6 forwarding for the case where packets must be fragmented at the 6LoWPAN sublayer. The first fragment is forwarded like any IPv6 packet and leaves a state in the intermediate hops to enable forwarding of the next fragments that do not have a IP header without the need to recompose the packet at every hop.

This can be broadly summarized in the following table:

```
+--------------------+------------+---------------------------------+
|   Forwarding Model |  Routing   |          Scheduling             |
+====================+============+=================================+
|G-MPLS Track Fwrding |     PCE    |Remote Monitoring and Schedule Mgt |
+--------------------+------------+---------------------------------+
|                    |            |   Static (Minimal Configuration)  |
+  classical IPv6    +     RPL    +---------------------------------+
|         /          |            |   Neighbor-to-Neighbor (SF0)      |
+ 6LoWPAN Fragment F. +------------+---------------------------------+
|                    |Reactive P2P|        Hop-by-Hop (TBD)           |
+--------------------+------------+---------------------------------+
```

Figure 2: Routing, Forwarding and Scheduling

3.5. A Non-Broadcast Multi-Access Radio Mesh Network

A 6TiSCH network is an IPv6 [RFC2460] subnet which, in its basic configuration, is a single Low Power Lossy Network (LLN) operating over a synchronized TSCH-based mesh.

Inside a 6TiSCH LLN, nodes rely on 6LoWPAN Header Compression (6LoWPAN HC) [RFC6282] to encode IPv6 packets. From the perspective of the network layer, 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.

6TiSCH nodes are not necessarily reachable from one another at Layer-2 and an LLN may span over multiple links. This effectively forms an homogeneous non-broadcast multi-access (NBMA) subnet, which is beyond the scope of existing IPv6 ND methods. Extensions to IPv6 ND have to be introduced.

Within that subnet, neighbor devices are discovered with 6LoWPAN Neighbor Discovery [RFC6775] (6LoWPAN ND), whereas RPL [RFC6550] enables routing in the so called Route Over fashion, either in storing (stateful) or non-storing (stateless, with routing headers) mode.

```
               ---+-------- ............ ------------
                  |      External Network       |
                  |                          +-----+
               +-----+                       | NME |
               |     | LLN Border            |     |
               |     | router                +-----+
               +-----+
             o    o   o
         o     o   o     o    o
        o   o 6LoWPAN + RPL o    o
            o   o   o       o
                    o  o
```

Figure 3: Basic Configuration of a 6TiSCH Network

6TiSCH nodes join the mesh by attaching to nodes that are already members of the mesh. Some nodes act as routers for 6LoWPAN ND and RPL operations, as detailed in Section 4.1. Security aspects of the join process by which a device obtains access to the network are discussed in Section 6.

With TSCH, devices are time-synchronized at the MAC level. The use of a particular RPL Instance for time synchronization is discussed in Section 4.2.4. With this mechanism, the time synchronization starts at the RPL root and follows the RPL DODAGs with no timing loop.

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 6TiSCH node, 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". RPL adds artifacts in the data packets that are compressed with a 6LoWPAN addition 6LoRH [I-D.ietf-roll-routing-dispatch].

Additional routing and scheduling protocols may be deployed to establish on-demand Peer-to-Peer routes with particular characteristics inside the 6TiSCH network. This may be achieved in a centralized fashion by a PCE [PCE] that programs both the routes and the schedules inside the 6TiSCH nodes, or by in a distributed fashion using a reactive routing protocol and a Hop-by-Hop scheduling protocol.

A Backbone Router may be connected to the node that acts as RPL root and / or 6LoWPAN 6LBR and provides connectivity to the larger campus / factory plant network over a high speed backbone or a back-haul

link. A Backbone Router may perform proxy IPv6 Neighbor Discovery (ND) [RFC4861] operations over the backbone on behalf of the 6TiSCH nodes so they can share a same IPv6 subnet and appear to be connected to the same backbone as classical devices. A Backbone Router 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].

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 architecture also expects that a 6LoWPAN node that is not aware at all of the RPL protocol may also connect as a host but the specifications for this to happen are not available at the time of this writing.

3.6. A Multi-Link Subnet Model

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 requires work to standardize the the registration of 6LoWPAN nodes to the Backbone Routers.

In the extended configuration, a Backbone Router (6BBR) operates as described in [I-D.ietf-6lo-backbone-router]. 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.

```
                  ---+-------- ............ ------------
                     |      External Network       |
                     |                          +-----+
                     |             +-----+      | NME |
                  +-----+          |  +-----+   |     |
                  |     | Router   |  | PCE |   +-----+
                  |     |          +--|     |
                  +-----+             +-----+
                     |                   |
                     | Subnet Backbone   |
               +--------------------+------------------+
               |                    |                  |
            +-----+             +-----+             +-----+
            |     | Backbone    |     | Backbone    |     | Backbone
       o    |     | router      |     | router      |     | router
            +-----+             +-----+             +-----+
       o                  o                   o                 o   o
           o    o   o         o   o  o   o         o  o   o    o
      o             o        o  LLN      o      o         o      o
         o   o    o      o      o o     o  o   o    o    o     o
```

Figure 4: Extended Configuration of a 6TiSCH Network

As detailed in Section 4.1 the 6LoWPAN ND 6LBR and the root of the RPL network need to be collocated and share information about the devices that is learned through either protocol but not both. The combined RPL root and 6LBR may be collocated with the 6BBR, or directly attached to the 6BBR. In the latter case, it leverages the extended registration process defined in [I-D.ietf-6lo-backbone-router] to proxy the 6LoWPAN ND registration to the 6BBR on behalf of the LLN nodes, so that the 6BBR may in turn perform proxy classical 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. The DetNet Architecture [I-D.finn-detnet-architecture] studies Layer-3 aspects of Deterministic Networks, and covers networks that span multiple Layer-2 domains.

3.7. Join Process and Registration

As detailed in Section 4.1 the combined 6LoWPAN ND 6LBR and root of the RPL network learn information such as the device Unique ID (from 6LoWPAN ND) and the updated Sequence Number (from RPL), and perform 6LoWPAN ND proxy registration to the 6BBR of behalf of the LLN nodes. Figure 5 illustrates the periodic signaling that starts at the leaf

node with 6LoWPAN ND, is then carried over RPL to the RPL root, and then 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.

```
 6LoWPAN Node        6LR             6LBR            6BBR
  (RPL leaf)       (router)         (root)
      |               |               |               |
      |  6LoWPAN ND   |6LoWPAN ND+RPL | Efficient ND  | IPv6 ND
      |   LLN link    |Route-Over mesh|  IPv6 link    | Backbone
      |               |               |               |
      |  NS(ARO)      |               |               |
      |-------------->|               |               |
      | 6LoWPAN ND    | DAR (then DAO)|               |
      |               |-------------->|               |
      |               |               |  NS(ARO)      |
      |               |               |-------------->|
      |               |               |               | DAD
      |               |               |               |------>
      |               |               |               |
      |               |               |  NA(ARO)      |
      |               |               |<--------------|
      |               | DAC           |               |
      |               |<--------------|               |
      |  NA(ARO)      |               |               |
      |<--------------|               |               |
```

Figure 5: (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.

3.8. Dependencies on Work In Progress

In order to control the complexity and the size of the 6TiSCH work, the architecture and the associated IETF work are staged and the WG is expected to recharter multiple times. This document is incremented as the work progresses following the evolution of the WG charter and the availability of dependent work. The intent is to publish when the WG concludes.

At the time of this writing:

- The architecture of the operation of RPL over a dynamic schedule is being studied at 6TISCH as the second iteration of the charter.

- The need of a reactive routing protocol to establish on-demand constraint-optimized routes and a reservation protocol to establish Layer-3 Tracks is being discussed at 6TiSCH but not chartered for.

- 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], and a number of those requirements are fulfilled in [I-D.ietf-6lo-backbone-router].

- The work on centralized Track computation is deferred to a subsequent iteration of the 6TiSCH charter. The idea at the time of this writing is that 6TiSCH will apply the concepts of Deterministic Networking on a Layer-3 network. The 6TiSCH Architecture should thus inherit from the DetNet [I-D.finn-detnet-architecture] architecture and thus depends on it. The Path Computation Element (PCE) should be a core component of that architecture. Around the PCE, a protocol such as an extension to a TEAS [TEAS] protocol will be required to expose the 6TiSCH node 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, as computed by the PCE, to the 6TiSCH nodes.

- The security model and in particular the join process are being discussed at 6lo and 6TiSCH. PANA is presented in Section 3.1 as a candidate of choice for the join process but alternatives are discussed. Work resulting from [ACE] could be considered as well. Related contributions are presented in Appendix A.

- 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. At the time of this writing, a revision of the IEEE802.15.4 standard is expected early 2016. That revision 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.

- Work is being proposed at IEEE (802.15.12 PAR) for an LLC that would logically include the 6top sublayer. The interaction with the 6top sublayer and the Scheduling Functions described in this document are yet to be defined.

- 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.

4. Deeper Dive

4.1. 6LoWPAN (and RPL)

4.1.1. 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 5.3 of [I-D.ietf-6lo-backbone-router], on the Extended Address Registration Option (EARO), 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 an overlay that matches the instanceID, for instance a VLANID.

This architecture leverages [I-D.ietf-6lo-backbone-router] that extends 6LoWPAN ND [RFC6775] to carry the counter as an abstract Transaction ID (TID).

4.1.2. RPL Root And 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. This architecture suggests to collocate these functions by default, in which case the discovery of the 6LBR is automatic for RPL leaves.

When 6LoWPAN ND is coupled with RPL, the 6LBR and RPL root functionalities are co-located in order that the address of the 6LBR be indicated by RPL DIO messages and to associate the unique ID from the DAR/DAC exchange with the state that is maintained by RPL. 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.

Even though the root of the RPL network is integrated with the 6LBR, it is logically separated from the Backbone Router (6BBR) that is used to connect the 6TiSCH LLN to the backbone. This way, the root has all information from 6LoWPAN ND and RPL about the LLN devices attached to it.

This architecture also expects that the root of the RPL network (proxy-)registers the 6TiSCH nodes on their behalf to the 6BBR, for whatever operation the 6BBR performs on the backbone, such as ND proxy, or redistribution in a routing protocol. This relies on an extension of the 6LoWPAN ND registration described in [I-D.ietf-6lo-backbone-router].

This model supports the movement of a 6TiSCH device across the Multi-Link Subnet, and allows the proxy registration of 6TiSCH nodes deep into the 6TiSCH LLN by the 6LBR / RPL root. This requires an alteration from [RFC6775] whereby the Target Address of the NS message is registered as opposed to the Source, which, in the case of a proxy registration, is that of the 6LBR / RPL root itself.

4.2. TSCH and 6top

4.2.1. 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.ietf-6tisch-6top-protocol]. 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 4.4.2.

The 6top data model and management interfaces are further discussed in Section 4.4.3.

4.2.1.1. Hard Cells

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.

4.2.1.2. Soft Cells

6top contains a monitoring process which monitors the performance of cells, and can move a cell in the TSCH schedule when it performs poorly. 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.

4.2.2. Scheduling Functions and the 6P protocol

In the case of soft cells, the cell management entity that controls the dynamic attribution of cells to adapt to the dynamics of variable rate flows is called a Scheduling Function (SF). There may be multiple SFs with more or less aggressive reaction to the dynamics of the network. The 6TiSCH 6top Scheduling Function Zero (SF0) [I-D.ietf-6tisch-6top-sf0] provides a simple scheduling function that can be used by default by devices that support dynamic scheduling of soft cells.

The SF may be seen as divided between an upper bandwidth adaptation logic that is not aware of the particular technology that is used to obtain and release bandwidth, and an underlying service that maps those needs in the actual technology, which means mapping the bandwidth onto cells in the case of TSCH.

```
+------------------------+         +------------------------+
|  Scheduling Function   |         |  Scheduling Function   |
|  Bandwidth adaptation  |         |  Bandwidth adaptation  |
+------------------------+         +------------------------+
|  Scheduling Function   |         |  Scheduling Function   |
| TSCH mapping to cells  |         | TSCH mapping to cells  |
+------------------------+         +------------------------+
| 6top cells negotiation | <- 6P -> | 6top cells negotiation |
+------------------------+         +------------------------+
        Device A                             Device B
```

Figure 6: SF/6P stack in 6top

The SF relies on 6top services that implement the 6top Protocol (6P) [I-D.ietf-6tisch-6top-protocol] to negotiate the precise cells that will be allocated or freed based on the schedule of the peer. It may be for instance that a peer wants to use a particular time slot that is free in its schedule, but that timeslot is already in use by the other peer for a communication with a third party on a different cell. The 6P protocol enables the peers to find an agreement in a transactional manner that ensures the final consistency of the nodes state.

4.2.3. 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.

Most OFs require metrics about reachability, such as the ETX. 6top creates and maintains an abstract neighbor table, and this state may be leveraged to feed an OF and/or store OF information as well. 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 IEEE802.15.4 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. Consideration was given towards finding a way to embed the Route Advertisements and the RPL DIO messages (both of which are multicast) into the IEEE802.15.4 Enhanced Beacons. It was determined that this produced undue timer coupling among layers, that the resulting packet size was potentially too large, and required it is not yet clear that there is any need for Enhanced Beacons in a production network.

4.2.4. 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. Optionally, RPL's periodic operations may be used to transport the network synchronization. This may mean that 6top would need to trigger (override) the trickle timer if no other traffic has occurred for such a time that nodes may get out of synchronization.

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 IEEE802.15.4 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.

4.2.5. 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 IEEE802.15.4 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 802.15.4 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.

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.

4.2.6. 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 abstraction of an IPv6 link is implemented as a pair of bundles of cells, one in each direction; 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.

```
             +-----+-----+-----+-----+-----+-----+-----+     +-----+
chan.Off. 0  |chnkA|chnkP|chnk7|chnkO|chnk2|chnkK|chnk1| ... |chnkZ|
             +-----+-----+-----+-----+-----+-----+-----+     +-----+
chan.Off. 1  |chnkB|chnkQ|chnkA|chnkP|chnk3|chnkL|chnk2| ... |chnk1|
             +-----+-----+-----+-----+-----+-----+-----+     +-----+
               ...
             +-----+-----+-----+-----+-----+-----+-----+     +-----+
chan.Off. 15 |chnkO|chnk6|chnkN|chnk1|chnkJ|chnkZ|chnkI| ... |chnkG|
             +-----+-----+-----+-----+-----+-----+-----+     +-----+
                0     1     2     3     4     5     6          M
```

Figure 7: 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.

4.3. 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 IEEE802.15.4 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 4.4. Link-layer frame forwarding interactions are discussed in Section 4.5, and Network-layer Packet routing is addressed in Section 4.6.

4.4. 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.

4.4.1. 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 is pre-established, for instance decided by a network administrator based on operational needs. It can be pre-configured into the nodes, or, more commonly, learned by a node when joining the network using standard IEEE802.15.4 Information Elements (IE). Regardless, the schedule remains unchanged after the node has joined a network. RPL is used on the resulting network. This "minimal" scheduling mechanism that implements this paradigm is detailed in [I-D.ietf-6tisch-minimal].

4.4.2. Neighbor-to-neighbor Scheduling

In the simplest instantiation of a 6TiSCH network described in Section 4.4.1, 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 traffic is dropped. If the size is configured to allow for peak emissions, energy is be wasted idle listening.

The 6top sublayer [I-D.wang-6tisch-6top-sublayer] defines a protocol for neighbor nodes to reserve soft cells to transmit 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.

In the most efficient instantiations 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 Scheduling Function (SF) such as SF0 [I-D.ietf-6tisch-6top-sf0] is used to monitor bandwidth usage and perform requests for dynamic allocation by the 6top sublayer. The SF 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.

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.

4.4.3. 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.

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). It is also 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.

With respect to Centralized routing and scheduling, the 6TiSCH Architecture is (expected to be) be an extension of the detnet work Deterministic Networking Architecture [I-D.finn-detnet-architecture], which studies Layer-3 aspects of Deterministic Networks, and covers networks that span multiple Layer-2 domains. The DetNet architecture is a form of SDN Architecture and is composed of three planes, a (User) Application Plane, a Controller Plane (where the PCE operates), and a Network Plane which in our case is the 6TiSCH LLN. The generic SDN architecture is discussed in Software-Defined Networking (SDN): Layers and Architecture Terminology [RFC7426] and is represented below:

           SDN Layers and Architecture Terminology per RFC 7426

```
                     o--------------------------------o
                     |                                |
                     | +-------------+   +----------+ |
                     | | Application |   |  Service | |
                     | +-------------+   +----------+ |
                     |       Application Plane        |
                     o---------------Y----------------o
                                     |
       *-----------------------------Y---------------------------------*
       |           Network Services Abstraction Layer (NSAL)           |
       *------Y------------------------------------------------Y-------*
              |                                                |
              |               Service Interface                |
              |                                                |
       o------Y------------------o       o---------------------Y------o
       |      |    Control Plane |       | Management Plane    |      |
       | +----Y----+   +-----+   |       |  +-----+       +----Y----+ |
       | | Service |   | App |   |       |  | App |       | Service | |
       | +----Y----+   +--Y--+   |       |  +--Y--+       +----Y----+ |
       |      |           |      |       |     |               |      |
       | *----Y-----------Y----* |       | *---Y---------------Y----* |
       | | Control Abstraction | |       | | Management Abstraction | |
       | |     Layer (CAL)     | |       | |      Layer (MAL)       | |
       | *----------Y----------* |       | *----------Y-------------* |
       |            |            |       |            |               |
       o------------|------------o       o------------|---------------o
                    |                                 |
                    | CP                              | MP
                    | Southbound                      | Southbound
                    | Interface                       | Interface
                    |                                 |
       *------------Y---------------------------------Y----------------*
       |         Device and resource Abstraction Layer (DAL)           |
       *------------Y---------------------------------Y----------------*
       |            |                                 |                |
       |    o-------Y----------o   +-----+   o--------Y----------o     |
       |    | Forwarding Plane |   | App |   | Operational Plane |     |
       |    o------------------o   +-----+   o-------------------o     |
       |                       Network Device                          |
       +---------------------------------------------------------------+
```

                                 Figure 8

   The PCE establishes end-to-end Tracks of hard cells, which are
   described in more details in Section 4.5.1. The DetNet work is
   expected to enable end to end Deterministic Path across heterogeneous

network (e.g. a 6TiSCH LLN and an Ethernet Backbone). This model fits the 6TiSCH extended configuration, whereby a 6BBR federates multiple 6TiSCH LLN in a single subnet over a backbone that can be, for instance, Ethernet or Wi-Fi. In that model, 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.

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. It is the responsibility of the PCE to compute a deterministic path and to end across the TSCH network and an IEEE802.1 TSN Ethernet backbone, and that of DetNet to enable end-to-end deterministic forwarding.

4.4.4. 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.

4.5. 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 4.6. 6TiSCH supports three different forwarding model, G-MPLS Track Forwarding (TF), 6LoWPAN Fragment Forwarding (FF) and IPv6 Forwarding (6F).

4.5.1. Track Forwarding

A Track is a directional 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 4.6.1. 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.

4.5.1.1. 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.

```
                      |                                    ^
   +--------------+   |                                    |
   |     IPv6     |   |                                    |
   +--------------+   |                                    |
   |  6LoWPAN HC  |   |                                    |
   +--------------+ ingress                              egress
   |     6top     |  sets     +----+          +----+     restores
   +--------------+ dmac to   |    |          |    |     dmac to
   |   TSCH MAC   |  brdcst   |    |          |    |      self
   +--------------+   |       |    |          |    |       |
   |   LLN PHY    |   +-------+    +--...-----+    +-------+
   +--------------+
```

Track Forwarding, Transport Mode

4.5.1.2. 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.

```
+--------------+
|     IPv6     |
+--------------+
|  6LoWPAN HC  |
+--------------+            set            restore
|     6top     |          +dmac+          +dmac+
+--------------+        to|brdcst       to|nexthop
|   TSCH MAC   |          |    |          |    |
+--------------+          |    |          |    |
|   LLN PHY    |    +-------+    +--...-----+    +-------+
+--------------+    |   ingress                 egress   |
                    |                                    |
+--------------+    |                                    |
|   LLN PHY    |    |                                    |
+--------------+    |                                    |
|   TSCH MAC   |    |                                    |
+--------------+    | dmac =                             | dmac =
|ISA100/WiHART |    | nexthop                            v nexthop
+--------------+
```

Figure 9: 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.

4.5.1.3. 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.

4.5.2. 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.

```
                         |                                    ^
      +--------------+   |                                    |
      |     IPv6     |   |       +----+          +----+       |
      +--------------+   |       |    |          |    |       |
      |  6LoWPAN HC  |   |       learn           learn        |
      +--------------+   |       |    |          |    |       |
      |     6top     |   |       |    |          |    |       |
      +--------------+   |       |    |          |    |       |
      |   TSCH MAC   |   |       |    |          |    |       |
      +--------------+   |       |    |          |    |       |
      |   LLN PHY    |   +-------+    +--...-----+    +-------+
      +--------------+
```

Figure 10: 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.

```
                         |                                    ^
      +--------------+   |                                    |
      |     IPv6     |   |                                    |
      +--------------+   |                                    |
      |  6LoWPAN HC  |   |       replay          replay       |
      +--------------+   |       |    |          |    |       |
      |     6top     |   |       |    |          |    |       |
      +--------------+   |       |    |          |    |       |
      |   TSCH MAC   |   |       |    |          |    |       |
      +--------------+   |       |    |          |    |       |
      |   LLN PHY    |   +-------+    +--...-----+    +-------+
      +--------------+
```

Figure 11: 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.

4.5.3. 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].

```
                      |                                    ^
   +--------------+   |                                    |
   |     IPv6     |   |       +-QoS+          +-QoS+       |
   +--------------+   |       |    |          |    |       |
   |  6LoWPAN HC  |   |       |    |          |    |       |
   +--------------+   |       |    |          |    |       |
   |     6top     |   |       |    |          |    |       |
   +--------------+   |       |    |          |    |       |
   |   TSCH MAC   |   |       |    |          |    |       |
   +--------------+   |       |    |          |    |       |
   |   LLN PHY    |   +-------+    +--...-----+    +-------+
   +--------------+
```

Figure 12: IP Forwarding

4.6. 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.

4.6.1. 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].

A compression mechanism for the RPL packet artifacts that integrates the compression of IP-in-IP encapsulation and the Routing Header type 3 [RFC6554] with that of the RPI in a 6LoWPAN dispatch/header type is concurrently being evaluated as [I-D.ietf-roll-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.

5. IANA Considerations

This specification does not require IANA action.

6. 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 6.1.

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.

6.1. Join Process Highlights

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

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

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

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 major activities:

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 13, 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.

```
   {joining node}     {neighbor}              {server, etc.}   Example:
   +---------+        +---------+                +---------+
   | Joining |        |  Join   |             +--|    CA   |certificate
   |  Node   |        |Assistant|             |  +---------+   issuance
   +---------+        +---------+             |  +---------+
      |                    |                  +--|Authoriz.| membership
      |<----Beaconing------|                  |  +---------+ test (JCE)
      |                    |                  |  +---------+
      |<--Authentication-->|                  +--| Routing | IP address
      |                    |<--Authorization-->|  +---------  assignment
      |<-------------------|                  |  +---------+
      |                    |                  +--| Gateway | backbone,
      |------------------->|                  |  +---------+    cloud
      |                    |<--Configuration-->|  +---------+
      |<-------------------|                  +--|Bandwidth|  PCE
                                                 +---------+  schedule
       .                     .                    .
       .                     .                    .
```

Figure 13: Network joining, with only authorization by third party

7. Acknowledgments

7.1. Contributors

The co-authors of this document are listed below:

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

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

Michael Richardson for his leadership role in the Security Design Team and his contribution throughout this document.

Rene Struik for the security section and his contribution to the Security Design Team.

Xavier Vilajosana who lead the design of the minimal support with RPL and contributed deeply to the 6top design and the G-MPLS operation of Track switching.

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

Thomas Watteyne for his contribution to the whole design, in particular on TSCH and security.

7.2. 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 SF0 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.

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

Also special thanks to Ralph Droms who performed the first INT Area Directorate review, that was very deep and through and radically changed the orientations of this document.

7.3. 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.

8. References

8.1. Normative References

[I-D.finn-detnet-architecture]
Finn, N., Thubert, P., and M. Teener, "Deterministic Networking Architecture", draft-finn-detnet-architecture-04 (work in progress), March 2016.

[I-D.ietf-6lo-backbone-router]
Thubert, P., "IPv6 Backbone Router", draft-ietf-6lo-backbone-router-01 (work in progress), March 2016.

[I-D.ietf-6tisch-minimal]
Vilajosana, X. and K. Pister, "Minimal 6TiSCH Configuration", draft-ietf-6tisch-minimal-15 (work in progress), February 2016.

[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-07 (work in progress), March 2016.

[I-D.ietf-roll-routing-dispatch]
Thubert, P., Bormann, C., Toutain, L., and R. Cragie, "6LoWPAN Routing Header", draft-ietf-roll-routing-dispatch-00 (work in progress), March 2016.

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <http://www.rfc-editor.org/info/rfc768>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <http://www.rfc-editor.org/info/rfc2460>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <http://www.rfc-editor.org/info/rfc4861>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <http://www.rfc-editor.org/info/rfc4862>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <http://www.rfc-editor.org/info/rfc6282>.

[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, <http://www.rfc-editor.org/info/rfc6550>.

[RFC6551] Vasseur, JP., Ed., Kim, M., Ed., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks", RFC 6551, DOI 10.17487/RFC6551, March 2012, <http://www.rfc-editor.org/info/rfc6551>.

[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, <http://www.rfc-editor.org/info/rfc6552>.

[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, DOI 10.17487/RFC6553, March 2012, <http://www.rfc-editor.org/info/rfc6553>.

[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, DOI 10.17487/RFC6554, March 2012, <http://www.rfc-editor.org/info/rfc6554>.

[RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <http://www.rfc-editor.org/info/rfc6775>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <http://www.rfc-editor.org/info/rfc7252>.

[RFC7554] Watteyne, T., Ed., Palattella, M., and L. Grieco, "Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement", RFC 7554, DOI 10.17487/RFC7554, May 2015, <http://www.rfc-editor.org/info/rfc7554>.

8.2. Informative References

[I-D.ietf-6tisch-6top-interface]
Wang, Q. and X. Vilajosana, "6TiSCH Operation Sublayer (6top) Interface", draft-ietf-6tisch-6top-interface-04 (work in progress), July 2015.

[I-D.ietf-6tisch-6top-protocol]
Wang, Q. and X. Vilajosana, "6top Protocol (6P)", draft-ietf-6tisch-6top-protocol-00 (work in progress), April 2016.

[I-D.ietf-6tisch-6top-sf0]
Dujovne, D., Grieco, L., Palattella, M., and N. Accettura, "6TiSCH 6top Scheduling Function Zero (SF0)", draft-ietf-6tisch-6top-sf0-00 (work in progress), May 2016.

[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-detnet-use-cases]
Grossman, E., Gunther, C., Thubert, P., Wetterwald, P., Raymond, J., Korhonen, J., Kaneko, Y., Das, S., Zha, Y., Varga, B., Farkas, J., Goetz, F., and J. Schmitt, "Deterministic Networking Use Cases", draft-ietf-detnet-use-cases-09 (work in progress), March 2016.

[I-D.ietf-manet-aodvv2]
Perkins, C., Ratliff, S., Dowdell, J., Steenbrink, L., and V. Mercieca, "Ad Hoc On-demand Distance Vector Version 2 (AODVv2) Routing", draft-ietf-manet-aodvv2-16 (work in progress), May 2016.

[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.struik-6tisch-security-architecture-elements]
Struik, R., Ohba, Y., and S. Das, "6TiSCH Security Architectural Elements, Desired Protocol Properties, and Framework", draft-struik-6tisch-security-architecture-elements-01 (work in progress), October 2014.

[I-D.svshah-tsvwg-deterministic-forwarding]
Shah, S. and P. Thubert, "Deterministic Forwarding PHB", draft-svshah-tsvwg-deterministic-forwarding-04 (work in progress), August 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-07 (work in progress), April 2016.

[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. and A. Bierman, "CoAP Management Interface", draft-vanderstok-core-comi-09 (work in progress), March 2016.

[I-D.wang-6tisch-6top-sublayer]
Wang, Q. and X. Vilajosana, "6TiSCH Operation Sublayer (6top)", draft-wang-6tisch-6top-sublayer-04 (work in progress), November 2015.

[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, DOI 10.17487/RFC2474, December 1998, <http://www.rfc-editor.org/info/rfc2474>.

[RFC2545] Marques, P. and F. Dupont, "Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing", RFC 2545, DOI 10.17487/RFC2545, March 1999, <http://www.rfc-editor.org/info/rfc2545>.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <http://www.rfc-editor.org/info/rfc3209>.

[RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", RFC 3444, DOI 10.17487/RFC3444, January 2003, <http://www.rfc-editor.org/info/rfc3444>.

[RFC3610] Whiting, D., Housley, R., and N. Ferguson, "Counter with CBC-MAC (CCM)", RFC 3610, DOI 10.17487/RFC3610, September 2003, <http://www.rfc-editor.org/info/rfc3610>.

[RFC3963] Devarapalli, V., Wakikawa, R., Petrescu, A., and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol", RFC 3963, DOI 10.17487/RFC3963, January 2005, <http://www.rfc-editor.org/info/rfc3963>.

[RFC3971] Arkko, J., Ed., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, DOI 10.17487/RFC3971, March 2005, <http://www.rfc-editor.org/info/rfc3971>.

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, DOI 10.17487/RFC3972, March 2005, <http://www.rfc-editor.org/info/rfc3972>.

[RFC4080] Hancock, R., Karagiannis, G., Loughney, J., and S. Van den Bosch, "Next Steps in Signaling (NSIS): Framework", RFC 4080, DOI 10.17487/RFC4080, June 2005, <http://www.rfc-editor.org/info/rfc4080>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <http://www.rfc-editor.org/info/rfc4291>.

[RFC4389] Thaler, D., Talwar, M., and C. Patel, "Neighbor Discovery Proxies (ND Proxy)", RFC 4389, DOI 10.17487/RFC4389, April 2006, <http://www.rfc-editor.org/info/rfc4389>.

[RFC4429] Moore, N., "Optimistic Duplicate Address Detection (DAD) for IPv6", RFC 4429, DOI 10.17487/RFC4429, April 2006, <http://www.rfc-editor.org/info/rfc4429>.

[RFC4903] Thaler, D., "Multi-Link Subnet Issues", RFC 4903, DOI 10.17487/RFC4903, June 2007, <http://www.rfc-editor.org/info/rfc4903>.

[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, DOI 10.17487/RFC4919, August 2007, <http://www.rfc-editor.org/info/rfc4919>.

[RFC5191] Forsberg, D., Ohba, Y., Ed., Patil, B., Tschofenig, H., and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA)", RFC 5191, DOI 10.17487/RFC5191, May 2008, <http://www.rfc-editor.org/info/rfc5191>.

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <http://www.rfc-editor.org/info/rfc5340>.

[RFC5889] Baccelli, E., Ed. and M. Townsley, Ed., "IP Addressing Model in Ad Hoc Networks", RFC 5889, DOI 10.17487/RFC5889, September 2010, <http://www.rfc-editor.org/info/rfc5889>.

[RFC5974] Manner, J., Karagiannis, G., and A. McDonald, "NSIS Signaling Layer Protocol (NSLP) for Quality-of-Service Signaling", RFC 5974, DOI 10.17487/RFC5974, October 2010, <http://www.rfc-editor.org/info/rfc5974>.

[RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, DOI 10.17487/RFC6275, July 2011, <http://www.rfc-editor.org/info/rfc6275>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <http://www.rfc-editor.org/info/rfc6347>.

[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, DOI 10.17487/RFC6620, May 2012, <http://www.rfc-editor.org/info/rfc6620>.

[RFC6655] McGrew, D. and D. Bailey, "AES-CCM Cipher Suites for Transport Layer Security (TLS)", RFC 6655, DOI 10.17487/RFC6655, July 2012, <http://www.rfc-editor.org/info/rfc6655>.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <http://www.rfc-editor.org/info/rfc6830>.

[RFC6997] Goyal, M., Ed., Baccelli, E., Philipp, M., Brandt, A., and J. Martocci, "Reactive Discovery of Point-to-Point Routes in Low-Power and Lossy Networks", RFC 6997, DOI 10.17487/RFC6997, August 2013, <http://www.rfc-editor.org/info/rfc6997>.

[RFC7426] Haleplidis, E., Ed., Pentikousis, K., Ed., Denazis, S., Hadi Salim, J., Meyer, D., and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", RFC 7426, DOI 10.17487/RFC7426, January 2015, <http://www.rfc-editor.org/info/rfc7426>.

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/>.

[DETNET] IETF, "Deterministic Networking", <https://datatracker.ietf.org/doc/charter-ietf-detnet/>.

[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".

[IEC62439] IEC, "Industrial communication networks - High availability automation networks - Part 3: Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR) - IEC62439-3", 2012, <https://webstore.iec.ch/publication/7018>.

[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".

[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.

Appendix A. Personal submissions relevant to upcoming work

This document covers a portion of the total work that is needed to cover the full 6TiSCH architecture. Missing portions at this time 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.struik-6tisch-security-architecture-elements] describes 6TiSCH security architectural elements with high level requirements and the security framework that are relevant for the design of the 6TiSCH security solution.

Author's Address

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