6TiSCH
Internet-Draft
Intended status: Informational
Expires: April 20, 2020

P. Thubert, Ed.
Cisco Systems

October 18, 2019

An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4
draft-ietf-6tisch-architecture-27

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 https://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on April 20, 2020.

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 range, and in circumstances where wiring may be impractical, for instance on fast-moving or rotating devices.

On the other hand, Deterministic Networking maximizes the packet delivery ratio within a bounded latency so as to enable mission-critical machine-to-machine (M2M) operations. Applications that need such networks are presented in [RFC8578]. The considered applications include Professional Media, Industrial Automation Control Systems (IACS), building automation, in-vehicle command and control, commercial automation and asset tracking with mobile scenarios, as well as gaming, drones and edge robotic control, and home automation applications.

The Timeslotted Channel Hopping (TSCH) [RFC7554] mode of the IEEE Std. 802.15.4 [IEEE802154] Medium Access Control (MAC) was introduced with the IEEE Std. 802.15.4e [IEEE802154e] amendment and is now retrofitted in the main standard. For all practical purposes, this document is expected to be insensitive to the revisions of that standard, which is thus referenced without a date. TSCH is both a Time-Division Multiplexing and a Frequency-Division Multiplexing technique whereby a different channel can be used for each

transmission, and that allows to schedule transmissions for deterministic operations, and applies to the slower and most energy constrained wireless use cases.

The scheduled operation provides for a more reliable experience which can be used to monitor and manage resources, e.g., energy and water, in a more efficient fashion.

Proven Deterministic Networking standards for use in Process Control, including ISA100.11a [ISA100.11a] and WirelessHART [WirelessHART], have demonstrated the capabilities of the IEEE Std. 802.15.4 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.

To enable the convergence of Information Technology (IT) and Operational Technology (OT) in Low-Power Lossy Networks (LLNs), the 6TiSCH Architecture supports an IETF suite of protocols over the IEEE Std. 802.15.4 TSCH MAC to provide IP connectivity for energy and otherwise constrained wireless devices.

The 6TiSCH Architecture relies on IPv6 [RFC8200] and the use of routing to provide large scaling capabilities. The addition of a high-speed federating backbone adds yet another degree of scalability to the design. The backbone is typically a Layer-2 transit Link such as an Ethernet bridged network, but it can also be a more complex routed structure.

The 6TiSCH Architecture introduces an IPv6 Multi-Link subnet model that is composed of a federating backbone and a number of IEEE Std. 802.15.4 TSCH low-power wireless networks federated and synchronized by Backbone Routers. If the backbone is a Layer-2 transit Link then the Backbone Routers can operate as an IPv6 Neighbor Discovery (IPv6 ND) [RFC4861] proxy.

The 6TiSCH Architecture leverages 6LoWPAN [RFC4944] to adapt IPv6 to the constrained media and RPL [RFC6550] for the distributed routing operations.

Centralized routing refers to a model where routes are computed and resources are allocated from a central controller. This is particularly helpful to schedule deterministic multihop transmissions. In contrast, Distributed Routing refers to a model that relies on concurrent peer to peer protocol exchanges for TSCH resource allocation and routing operations.

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 highly scalable solutions such as used in Advanced Metering Infrastructure [AMI] solutions that leverage distributed routing to enable multipath forwarding over large LLN meshes.

2. Terminology

2.1. New Terms

The draft does not reuse terms from the IEEE Std. 802.15.4 [IEEE802154] standard such as "path" or "link" which bear a meaning that is quite different from classical IETF parlance.

This document adds the following terms:

6TiSCH (IPv6 over the TSCH mode of IEEE 802.15.4): 6TiSCH defines an adaptation sublayer for IPv6 over TSCH called 6top, a set of protocols for setting up a TSCH schedule in distributed approach, and a security solution. 6TiSCH may be extended in the future for other MAC/PHY pairs providing a service similar to TSCH.

6top (6TiSCH Operation Sublayer): The next higher layer of the IEEE Std. 802.15.4 TSCH MAC layer. 6top provides the abstraction of an IP link over a TSCH MAC, schedules packets over TSCH cells, and exposes a management interface to schedule TSCH cells.

6P (6top Protocol): The protocol defined in [RFC8480]. 6P enables Layer-2 peers to allocate, move or deallocate cells in their respective schedules to communicate. 6P operates at the 6top layer.

6P Transaction: A 2-way or 3-way sequence of 6P messages used by Layer-2 peers to modify their communication schedule.

ASN (Absolute Slot Number): Defined in [IEEE802154], the ASN is the total number of timeslots that have elapsed since the Epoch Time when the TSCH network started. Incremented by one at each timeslot. It is wide enough to not roll over in practice.

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

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

Layer-2 vs. Layer-3 bundle: Bundles are associated for either Layer-2 (switching) or Layer-3 (routing) forwarding operations. A pair of Layer-3 bundles (one for each direction) maps to an IP Link with a neighbor, whereas a set of Layer-2 bundles (of an "arbitrary" cardinality and direction) corresponds to the relation of one or more incoming bundle(s) from the previous-hop neighbor(s) with one or more outgoing bundle(s) to the next-hop neighbor(s) along a Track as part of the switching role, which may include replication and elimination.

CCA (Clear Channel Assessment): A mechanism defined in [IEEE802154] whereby nodes listen to the channel before sending to detect ongoing transmissions from other parties. Because the network is synchronized, CCA cannot be used to detect colliding transmissions within the same network, but it can be used to detect other radio networks in vicinity.

cell: A unit of transmission resource in the CDU matrix, a cell is identified by a slotOffset and a channelOffset. A cell can be scheduled or unscheduled.

Channel Distribution/Usage (CDU) matrix: : A matrix of cells (i,j) representing the spectrum (channel) distribution among the different nodes in the 6TiSCH network. The CDU matrix has width in timeslots, equal to the period of the network scheduling operation, and height equal to the number of available channels. Every cell (i,j) in the CDU, identified by (slotOffset, channelOffset), belongs to a specific chunk.

channelOffset: Identifies a row in the TSCH schedule. The number of channelOffset values is bounded by the number of available frequencies. The channelOffset translates into a frequency with a function that depends on the absolute time when the communication takes place, resulting in a channel hopping operation.

chunk: A well-known list of cells, distributed in time and frequency, within a CDU matrix. A chunk represents a portion of a CDU matrix. The partition of the CDU matrix in chunks is globally known by all the nodes in the network to support the appropriation process, which is a

negotiation between nodes within an interference domain. A node that manages to appropriate a chunk gets to decide which transmissions will occur over the cells in the chunk within its interference domain, i.e., a parent node will decide when the cells within the appropriated chunk are used and by which node, among its children.

CoJP (Constrained Join Protocol): The Constrained Join Protocol (CoJP) enables a pledge to securely join a 6TiSCH network and obtain network parameters over a secure channel. Minimal Security Framework for 6TiSCH [I-D.ietf-6tisch-minimal-security] defines the minimal CoJP setup with pre-shared keys defined. In that mode, CoJP can operate with a single round trip exchange.

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

deterministic network: The generic concept of deterministic network is defined in [I-D.ietf-detnet-architecture]. When applied to 6TiSCH, it refers to the reservation of Tracks which guarantees an end-to-end latency and optimizes the Packet Delivery Ratio (PDR) for well-characterized flows.

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

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

EB (Enhanced Beacon): A special frame defined in [IEEE802154] used by a node, including the JP, to announce the presence of the network. It contains enough information for a pledge to synchronize to the network.

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

hopping sequence: Ordered sequence of frequencies, identified by a Hopping_Sequence_ID, used for channel hopping when translating the channelOffset value into a frequency.

IE (Information Element): Type-Length-Value containers placed at the end of the MAC header, used to pass data between layers or devices. Some IE identifiers are managed by the IEEE [IEEE802154]. Some IE identifiers are managed by the IETF [RFC8137], and

[I-D.ietf-6tisch-enrollment-enhanced-beacon] uses one subtype to support the selection of the Join Proxy.

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

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

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

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

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

link: A communication facility or medium over which nodes can communicate at the Link-Layer, the layer immediately below IP. In 6TiSCH, the concept is implemented as a collection of Layer-3 bundles. Note: the IETF parlance for the term "Link" is adopted, as opposed to the IEEE Std. 802.15.4 terminology.

Operational Technology: OT refers to technology used in automation, for instance in industrial control networks. The convergence of IT and OT is the main object of the Industrial Internet of Things (IIOT).

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

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

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

scheduled cell: A cell which is assigned a neighbor MAC address (broadcast address is also possible), and one or more of the following flags: TX, RX, Shared and Timekeeping. A

scheduled cell can be used by the IEEE Std. 802.15.4 TSCH implementation to communicate. A scheduled cell can either be a hard or a soft cell.

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

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

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

slotframe: A collection of timeslots repeating in time, analogous to a superframe in that it defines periods of communication opportunities. It is characterized by a slotframe_ID, and a slotframe_size. Multiple slotframes can coexist in a node's schedule, i.e., a node can have multiple activities scheduled in different slotframes, based on the priority of its packets/traffic flows. The timeslots in the Slotframe are indexed by the SlotOffset; the first timeslot is at SlotOffset 0.

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

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

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

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

Track: A Track is a Directed Acyclic Graph (DAG) that is used as a complex multi-hop path to the destination(s) of the path. In the case of unicast traffic, the Track is a Destination Oriented DAG (DODAG) where the Root of the DODAG is the destination of the unicast traffic. A Track enables replication, elimination and reordering functions on the way (more on those functions in the Deterministic

Networking Architecture [I-D.ietf-detnet-architecture]). A Track reservation locks physical resources such as cells and buffers in every node along the DODAG. A Track is associated with a owner that can be for instance the destination of the Track.

TrackID: A TrackID is either globally unique, or locally unique to the Track owner, in which case the identification of the owner must be provided together with the TrackID to provide a full reference to the Track. If the Track owner is the destination of the Track then the destination IP address of packets along the Track can be used as identification of the owner and a local InstanceID [RFC6550] can be used as TrackID. In that case, a RPL Packet Information [RFC6550] in an IPv6 packet can unambiguously identify the Track and can be expressed in a compressed form using [RFC8138].

TSCH: A medium access mode of the IEEE Std. 802.15.4 [IEEE802154] standard which uses time synchronization to achieve ultra-low-power operation, and channel hopping to enable high reliability.

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

Unscheduled Cell: A cell which is not used by the IEEE Std. 802.15.4 TSCH implementation.

2.2. Abbreviations

This document uses the following abbreviations:

6BBR: 6LoWPAN Backbone Router (router with a proxy ND function)

6LBR: 6LoWPAN Border Router (authoritative on DAD)

6LN: 6LoWPAN Node

6LR: 6LoWPAN Router (relay to the registration process)

6CIO: Capability Indication Option

(E)ARO: (Extended) Address Registration Option

(E)DAR: (Extended) Duplicate Address Request

(E)DAC: (Extended) Duplicate Address Confirmation

DAD: Duplicate Address Detection

DODAG: Destination-Oriented Directed Acyclic Graph

LLN: Low-Power and Lossy Network (a typical IoT network)

NA: Neighbor Advertisement

NCE: Neighbor Cache Entry

ND: Neighbor Discovery

NDP: Neighbor Discovery Protocol

PCE: Path Computation Element

NME: Network Management Entity

ROVR: Registration Ownership Verifier (pronounced rover)

RPL: IPv6 Routing Protocol for LLNs (pronounced ripple)

RA: Router Advertisement

RS: Router Solicitation

TSCH: timeslotted Channel Hopping

TID: Transaction ID (a sequence counter in the EARO)

2.3. Related Documents

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

The draft uses domain-specific terminology defined or referenced in:

6LoWPAN ND "Neighbor Discovery Optimization for Low-power and Lossy Networks" [RFC6775] and "Registration Extensions for 6LoWPAN Neighbor Discovery" [RFC8505],

"Terms Used in Routing for Low-Power and Lossy Networks (LLNs)" [RFC7102],

and RPL "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)" [RFC6552], and "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks" [RFC6550].

Other terms in use in LLNs are found in "Terminology for Constrained-Node Networks" [RFC7228].

Readers are expected to be familiar with all the terms and concepts that are discussed in

- "Neighbor Discovery for IP version 6" [RFC4861], and "IPv6 Stateless Address Autoconfiguration" [RFC4862].

In addition, readers would benefit from reading:

- "Problem Statement and Requirements for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing" [RFC6606],
- "Multi-Link Subnet Issues" [RFC4903], and
- "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals" [RFC4919]

prior to this specification for a clear understanding of the art in ND-proxying and binding.

3. High Level Architecture

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

A 6TiSCH network is an IPv6 [RFC8200] subnet which, in its basic configuration illustrated in Figure 1, is a single Low-Power Lossy Network (LLN) operating over a synchronized TSCH-based mesh.

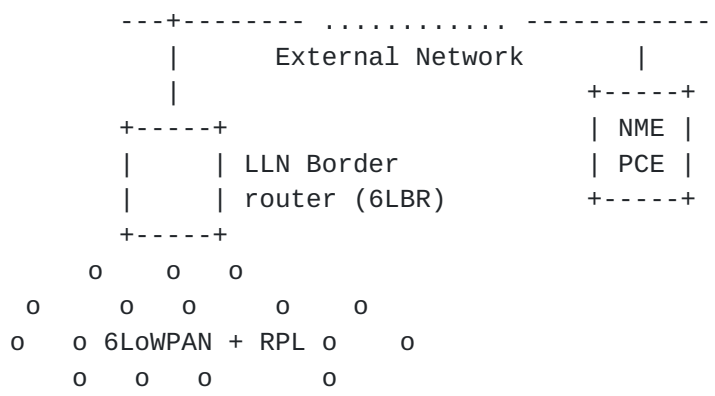

Figure 1: Basic Configuration of a 6TiSCH Network

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 IEEE Std. 802.15.4-compliant radio) may be seen as a collection of Links with different capabilities for unicast or multicast services.

6TiSCH nodes join a mesh network by attaching to nodes that are already members of the mesh (see Section 4.2.1). The security aspects of the join process are further detailed in Section 6. In a mesh network, 6TiSCH nodes are not necessarily reachable from one another at Layer-2 and an LLN may span over multiple links.

This forms a homogeneous non-broadcast multi-access (NBMA) subnet, which is beyond the scope of IPv6 Neighbor Discovery (IPv6 ND) [RFC4861][RFC4862]. 6LoWPAN Neighbor Discovery (6LoWPAN ND) [RFC6775][RFC8505] specifies extensions to IPv6 ND that enable ND operations in this type of subnet that can be protected against address theft and impersonation with [I-D.ietf-6lo-ap-nd].

Once it has joined the 6TiSCH network, a node acquires IPv6 Addresses and register them using 6LoWPAN ND. This guarantees that the addresses are unique and protects the address ownership over the subnet, more in Section 4.2.2.

Within the NBMA subnet, RPL [RFC6550] enables routing in the so-called Route Over fashion, either in storing (stateful) or non-storing (stateless, with routing headers) mode. From there, some nodes can act as routers for 6LoWPAN ND and RPL operations, as detailed in Section 4.1.

With TSCH, devices are time-synchronized at the MAC level. The use of a particular RPL Instance for time synchronization is discussed in Section 4.3.4. With this mechanism, the time synchronization starts at the RPL Root and follows the RPL loopless routing topology.

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 (6LBR), acts as RPL Root, 6LoWPAN HC terminator, and Border Router for the LLN to the outside. The 6LBR 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 [RFC8138].

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 Path Computation Element (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.

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 described in [I-D.ietf-roll-unaware-leaves].

3.2. A Multi-Link Subnet Model

An extended configuration of the subnet comprises multiple LLNs as illustrated in Figure 2. In the extended configuration, a Routing Registrar [RFC8505] 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. The Routing registrar may perform IPv6 ND proxy operations, or redistribute the registration in a routing protocol such as OSPF [RFC5340] or BGP [RFC2545], or inject a route in a mobility protocol such as MIPv6 [RFC6275], NEMO [RFC3963], or LISP [RFC6830].

Multiple LLNs can be interconnected and possibly synchronized over a backbone, which can be wired or wireless. The backbone can operate with IPv6 ND [RFC4861][RFC4862] procedures or an hybrid of IPv6 ND and 6LoWPAN ND [RFC6775][RFC8505][I-D.ietf-6lo-ap-nd].

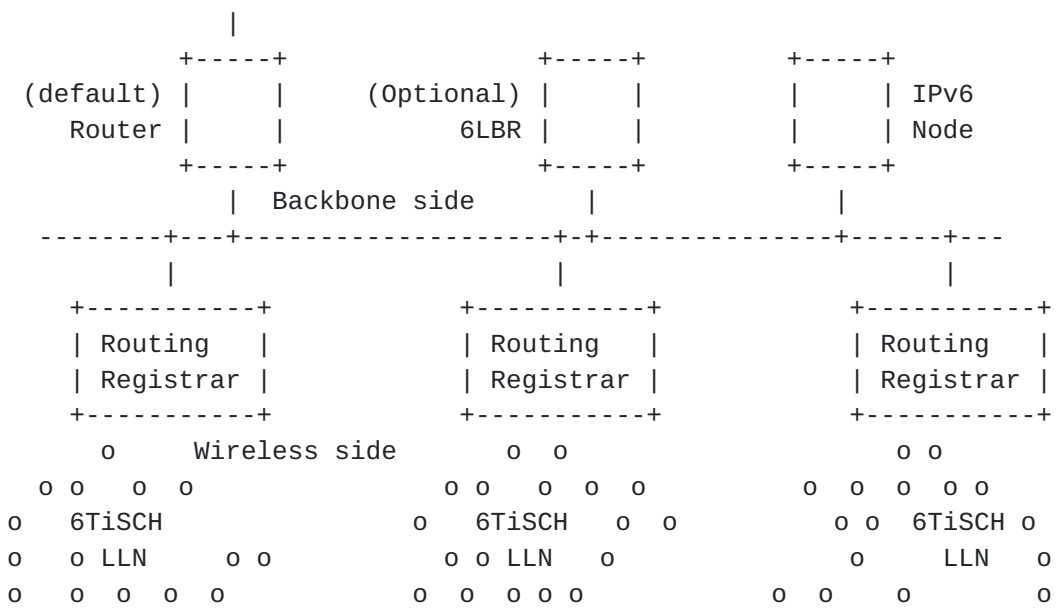

Figure 2: Extended Configuration of a 6TiSCH Network

A Routing Registrar that performs proxy IPv6 ND operations over the backbone on behalf of the 6TiSCH nodes is called a Backbone Router (6BBR) [I-D.ietf-6lo-backbone-router]. The 6BBRs are placed along the wireless edge of a Backbone, and federate multiple wireless links to form a single MultiLink Subnet. The 6BBRs synchronize with one another over the backbone, so as to ensure that the multiple LLNs that form the IPv6 subnet stay tightly synchronized.

The use of multicast can also be reduced on the backbone with a registrar that would contribute to Duplicate Address Detection as well as Address Lookup using only unicast request/response exchanges. [I-D.thubert-6man-unicast-lookup] is a proposed method that presents an example of how to this could be achieved with an extension of [RFC8505], using an optional 6LBR as a SubNet-level registrar, as illustrated in Figure 2.

As detailed in Section 4.1 the 6LBR that serves the LLN and the Root of the RPL network need to share information about the devices that are learned through either 6LoWPAN ND or RPL but not both. The preferred way of achieving this is to collocate/combine them. 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 [RFC8505] 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.

The DetNet Architecture [I-D.ietf-detnet-architecture] studies Layer-3 aspects of Deterministic Networks, and covers networks that span multiple Layer-2 domains. 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.

3.3. TSCH: A Deterministic MAC Layer

Though at a different time scale (several orders of magnitude), both IEEE Std. 802.1TSN and IEEE Std. 802.15.4 TSCH 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 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 co-channel interference.

6TiSCH builds on the IEEE Std. 802.15.4 TSCH 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.ietf-detnet-architecture], and IETF components such as the 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 and mobile workers accessing local devices.

3.4. 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. The CDU matrix can be formatted in chunks that can be allocated exclusively to particular nodes to enable distributed scheduling without collision. More in Section 4.3.5.

From the standpoint of a 6TiSCH node (at the MAC layer), its schedule is the collection of the timeslots 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 collide and require a device to wake up at a same time, in which case the slotframe with the highest priority is actionable.

The 6top sublayer (see Section 4.3 for more) hides the complexity of the schedule from the upper layers. The Link abstraction that IP traffic utilizes is composed of a pair of Layer-3 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.

Scheduling 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 either transmit or receive on one channel at any point of time. Scheduled cells that fulfil the same role, e.g., receive IP packets from a peer, are grouped in bundles.

The 6TiSCH architecture identifies four ways a schedule can be managed and CDU cells can be allocated: Static Scheduling, Neighbor-to-Neighbor Scheduling, Centralized (or 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 [S-ALOHA] fashion. The static schedule is distributed through the native methods in the TSCH MAC layer and does not preclude other scheduling operations to co-exist on a same 6TiSCH network. A static schedule is necessary for basic operations such as the join process and for interoperability during the network formation, which is specified as part of the Minimal 6TiSCH Configuration [RFC8180].

Neighbor-to-Neighbor Scheduling: This refers to the dynamic adaptation of the bandwidth of the Links that are used for IPv6 traffic between adjacent peers. Scheduling Functions such as the

"6TiSCH Minimal Scheduling Function (MSF)" [I-D.ietf-6tisch-msf] influence the operation of the MAC layer to add, update and remove cells in its own, and its peer's schedules using 6P [RFC8480], for the negotiation of the MAC resources.

Centralized (or 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 to minimize the interaction with and the load on the constrained device. This model is the TSCH adaption of the "DetNet Architecture" [I-D.ietf-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.

3.5. Distributed vs. Centralized Routing

6TiSCH enables a mixed model of centralized routes and distributed routes. Centralized routes can for example be computed by an entity such as a PCE. 6TiSCH leverages the RPL [RFC6550] routing protocol for interoperable distributed routing operations.

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 RPLInstanceID, in a format that reuses the RPLInstanceID as defined in RPL.

RPL [RFC6550] 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 [RFC6550], such as the Asymmetric AODV-P2P-RPL in Low-Power and Lossy Networks" [I-D.ietf-roll-aodv-rpl] AODV-RPL), which derives from the Ad Hoc On-demand Distance Vector Routing

(AODV) [I-D.ietf-manet-aodvv2] will be selected for Hop-by-hop Scheduling.

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.

3.6. Forwarding Over TSCH

The 6TiSCH architecture supports three different forwarding models. One is 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. The second derives from Generic MPLS (G-MPLS) for so-called Track Forwarding, whereby a frame received at a particular timeslot can be switched into another timeslot at Layer-2 without regard to the upper layer protocol. The third model is the 6LoWPAN Fragment Forwarding, which allows to forward individual 6loWPAN fragments along a route that is setup by the first fragment.

In more details:

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. Besides 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 DAG shapes that are more complex than a simple direct sequence of nodes.

6LoWPAN Fragment Forwarding: This is a 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.

A deeper dive on these operations can be found in Section 4.6.

The following table summarizes how the forwarding models apply to the various routing and scheduling possibilities:

Forwarding Model	Routing	Scheduling
classical IPv6 / 6LoWPAN Fragment	RPL	Static (Minimal Configuration)
		Neighbor-to-Neighbor (SF+6P)
	Reactive	Hop-by-Hop (AODV-RPL)
G-MPLS Track Fwding	PCE	Remote Monitoring and Schedule Mgt

3.7. 6TiSCH Stack

The IETF proposes multiple techniques for implementing functions related to routing, transport or security.

The 6TiSCH architecture limits the possible variations of the stack and recommends a number of base elements for LLN applications to control the complexity of possible deployments and device interactions, and to limit the size of the resulting object code. In particular, UDP [RFC0768], IPv6 [RFC8200] 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 protocol stack is represented in Figure 4:

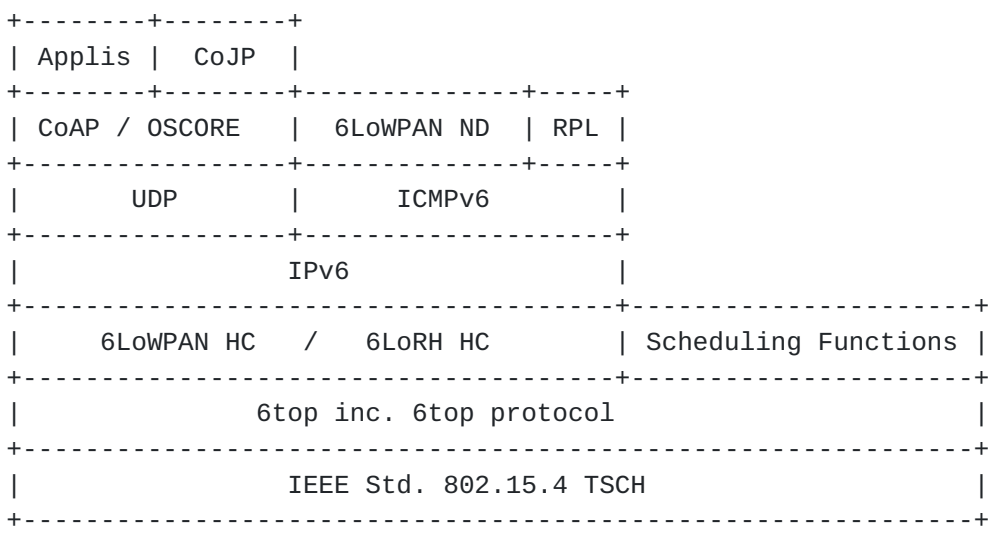

Figure 4: 6TiSCH Protocol Stack

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 [RFC8180] describes the operation of RPL over a static schedule used in a Slotted ALOHA fashion [S-ALOHA], 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 (6LoRH) [RFC8138] 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. "When to use RFC 6553, 6554 and IPv6-in-IPv6" [I-D.ietf-roll-useofrplinfo] provides the details on when headers or encapsulation are needed.

The Object Security for Constrained RESTful Environments (OSCORE) [I-D.ietf-core-object-security], is leveraged by the Constrained Join Protocol (CoJP) and is expected to be the primary protocol for the protection of the application payload as well. The application payload may also be protected by the Datagram Transport Layer Security (DTLS) [RFC6347] sitting either under CoAP or over CoAP so it can traverse proxies.

The 6TiSCH Operation sublayer (6top) 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, providing in particular dynamic cell allocation with the 6top Protocol (6P) [RFC8480].

The reference stack presented in this document was 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.

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.

3.8. Communication Paradigms and Interaction Models

Section 2.1 provides the terms of Communication Paradigms and Interaction Models, in relation with "On the Difference between Information Models and Data Models" [RFC3444]. A Communication Paradigm 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 Model would be more refined and could point to 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 - one sender, two receivers for redundancy - and its N-cast generalization are also provided. Those paradigms are frequently used in industrial automation, which is a major use case for IEEE Std. 802.15.4 TSCH wireless networks with [ISA100.11a] and [WirelessHART], that provides a wireless access to [HART] applications and devices.

This document 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 (multihop along a Track), and Application-layer (remote control) are discussed in Section 4.4. Link-Layer frame forwarding interactions are discussed in Section 4.6, and Network-layer Packet routing is addressed in Section 4.7.

4. Architecture Components

4.1. 6LoWPAN (and RPL)

A RPL DODAG is formed of a Root, a collection of routers, and leaves that are hosts. Hosts are nodes which do not forward packets that they did not generate. RPL-aware leaves will participate to RPL to advertise their own addresses, whereas RPL-unaware leaves depend on a connected RPL router to do so. RPL interacts with 6LoWPAN ND at multiple levels, in particular at the Root and in the RPL-unaware leaves.

4.1.1. RPL-Unaware Leaves and 6LoWPAN ND

RPL needs a set of information to advertise a leaf node through a Destination Advertisement Object (DAO) message and establish reachability.

"Routing for RPL Leaves" [I-D.ietf-roll-unaware-leaves] details the basic interaction of 6LoWPAN ND and RPL and enables a plain 6LN that supports [RFC8505] to obtain return connectivity via the RPL network as an RPL-unaware leaf. The leaf indicates that it requires reachability services for the Registered Address from a Routing Registrar by setting a 'R' flag in the Extended Address Registration Option [RFC8505], and it provides a TID that maps to a sequence number in section 7 of RPL [RFC6550].

[I-D.ietf-roll-unaware-leaves] also enables the leaf to signal the RPL InstanceID that it wants to participate to using the Opaque field of the EARO. On the backbone, the InstanceID is expected to be mapped to an overlay that matches the RPL Instance, e.g., a Virtual LAN (VLAN) or a virtual routing and forwarding (VRF) instance.

Though at the time of this writing the above specification enables a model where the separation is possible, this architecture recommends to collocate the functions of 6LBR and RPL Root.

4.1.2. 6LBR and RPL Root

With the 6LowPAN ND [RFC6775], information on the 6LBR is disseminated via an Authoritative Border Router Option (ABRO) in RA messages. [RFC8505] extends [RFC6775] to enable a registration for routing and proxy ND. The capability to support [RFC8505] is indicated in the 6LoWPAN Capability Indication Option (6CIO). The discovery and liveliness of the RPL Root are obtained through RPL [RFC6550] itself.

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 EDAR/EDAC [RFC8505] exchange with the state that is maintained by RPL.

Section 7 of [I-D.ietf-roll-unaware-leaves] specifies how the DAO messages are used to reconfirm the registration, thus eliminating a duplication of functionality between DAO and EDAR/EDAC messages, as illustrated in Figure 7. [I-D.ietf-roll-unaware-leaves] also provides the protocol elements that are needed when the 6LBR and RPL Root functionalities are not co-located.

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 is why in [RFC8505] the Registered Address is signaled in the Target Address field of the NS message as opposed to the IPv6 Source Address, which, in the case of a proxy registration, is that of the 6LBR / RPL Root itself.

4.2. Network Access and Addressing

4.2.1. Join Process

A new device, called the pledge, undergoes the join protocol to become a node in a 6TiSCH network. This usually occurs only once when the device is first powered on. The pledge communicates with the Join Registrar/Coordinator (JRC) of the network through a Join Proxy (JP), a radio neighbor of the pledge.

The JP is discovered though MAC layer beacons. When multiple JPs from possibly multiple networks are visible, trial and error till an acceptable position in the right network is obtained becomes ineffficient. [I-D.ietf-6tisch-enrollment-enhanced-beacon] adds a new subtype in the Information Element that was delegated to the IETF

[RFC8137] and provides visibility on the network that can be joined and the willingness by the JP and the Root to be used by the pledge.

The join protocol provides the following functionality:

- Mutual authentication
- Authorization
- Parameter distribution to the pledge over a secure channel

Minimal Security Framework for 6TiSCH [I-D.ietf-6tisch-minimal-security] defines the minimal mechanisms required for this join process to occur in a secure manner. The specification defines the Constrained Join Protocol (CoJP) that is used to distribute the parameters to the pledge over a secure session established through OSCORE [I-D.ietf-core-object-security], and a secure configuration of the network stack. In the minimal setting with pre-shared keys (PSKs), CoJP allows the pledge to join after a single round-trip exchange with the JRC. The provisioning of the PSK to the pledge and the JRC needs to be done out of band, through a 'one-touch' bootstrapping process, which effectively enrolls the pledge into the domain managed by the JRC.

In certain use cases, the 'one touch' bootstrapping is not feasible due to the operational constraints and the enrollment of the pledge into the domain needs to occur in-band. This is handled through a 'zero-touch' extension of the Minimal Security Framework for 6TiSCH. Zero touch [I-D.ietf-6tisch-dtsecurity-zerotouch-join] extension leverages the 'Bootstrapping Remote Secure Key Infrastructures (BRSKI)' [[I-D.ietf-anima-bootstrapping-keyinfra] work to establish a shared secret between a pledge and the JRC without necessarily having them belong to a common (security) domain at join time. This happens through inter-domain communication occurring between the JRC of the network and the domain of the pledge, represented by a fourth entity, Manufacturer Authorized Signing Authority (MASA). Once the zero-touch exchange completes, the CoJP exchange defined in [I-D.ietf-6tisch-minimal-security] is carried over the secure session established between the pledge and the JRC.

Figure 5 depicts the join process and where a Link-Local Address (LLA) is used, versus a Global Unicast Address (GUA).

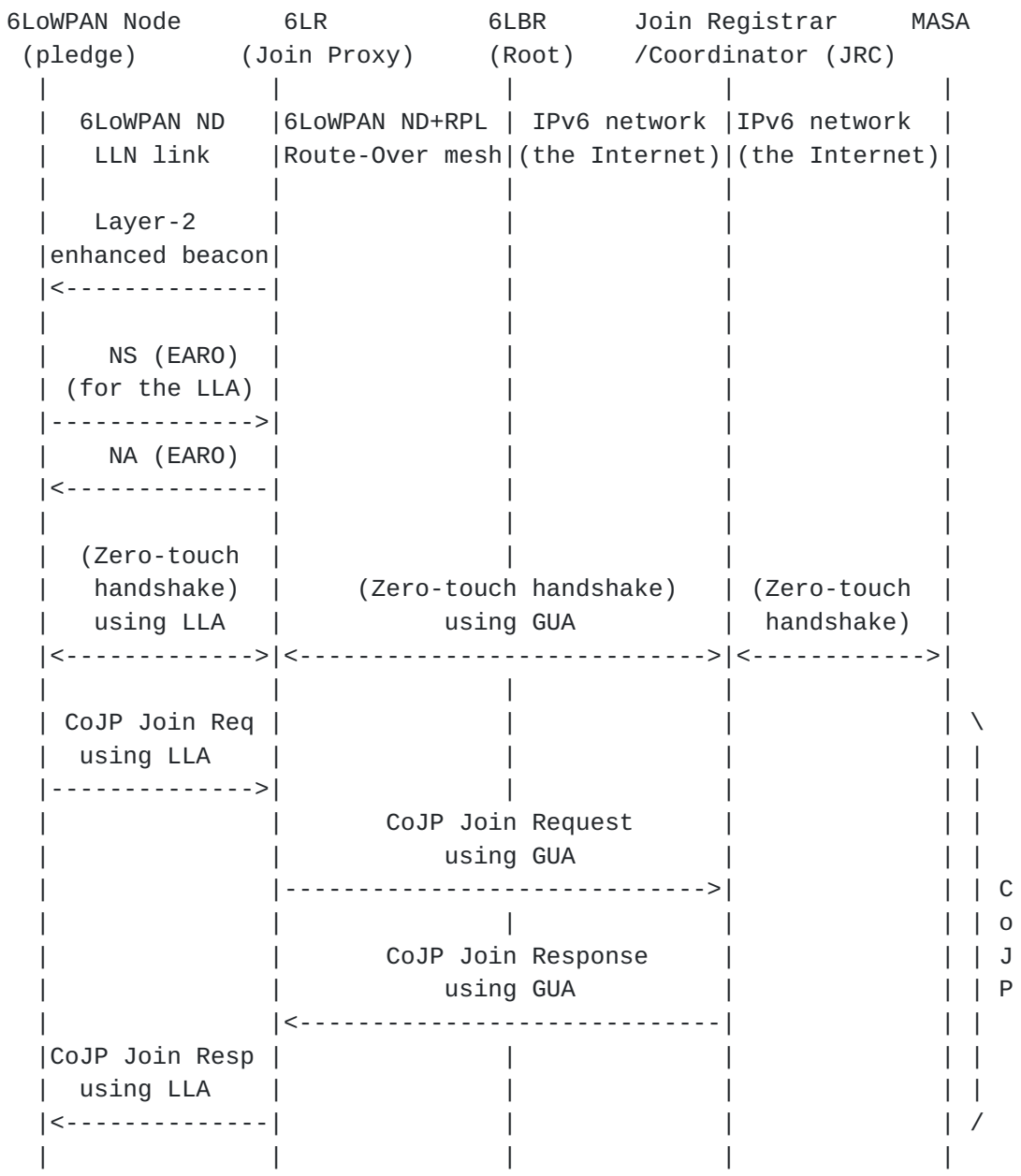

Figure 5: Join process in a Multi-Link Subnet. Parentheses () denote optional exchanges.

4.2.2. Registration

Once the pledge successfully completes the CoJP protocol and becomes a network node, it obtains the network prefix from neighboring routers and registers its IPv6 addresses. 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 6 illustrates the initial IPv6 signaling that enables a 6LN to form a global address and register it to a 6LBR using 6LoWPAN ND [RFC8505], is then carried over RPL to the RPL Root, and then to the 6BBR. This flow happens just once when the address is created and first registered.

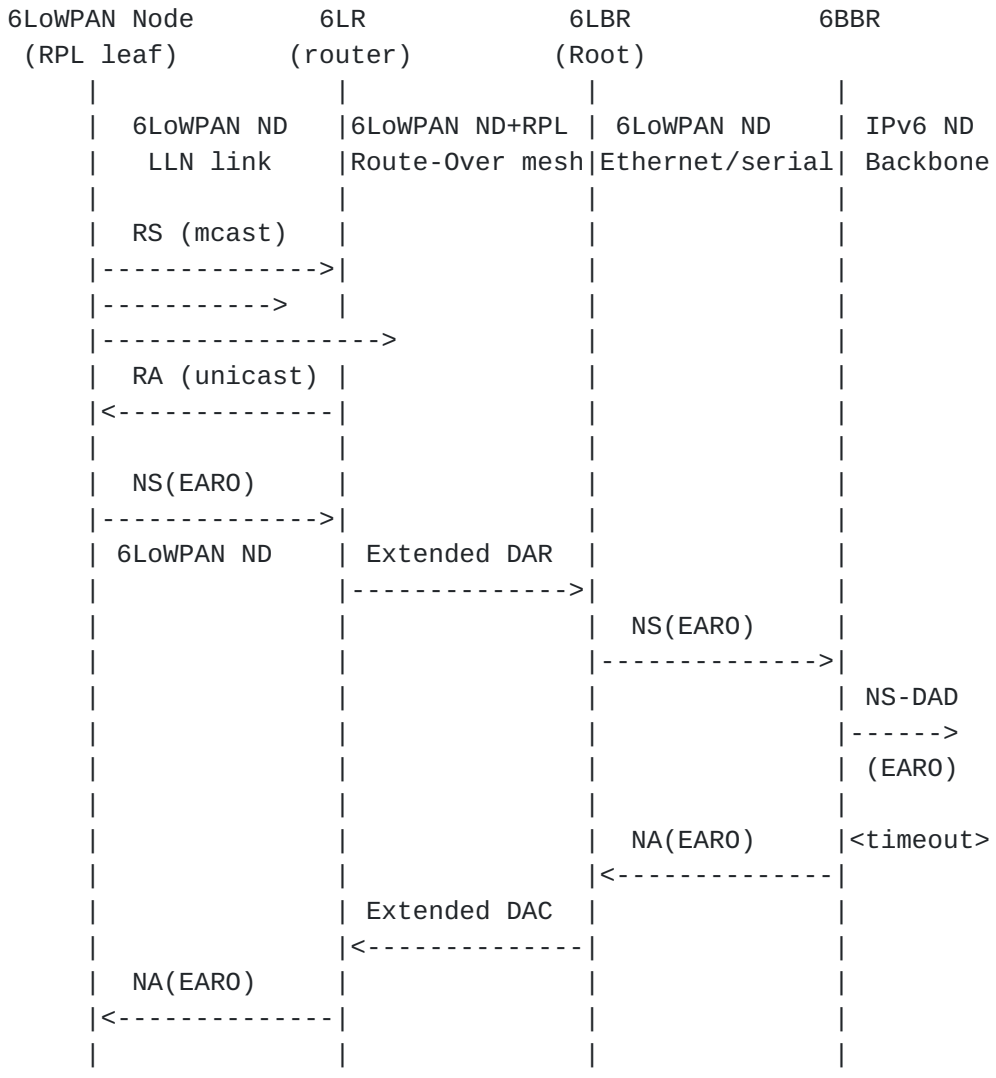

Figure 6: Initial Registration Flow over Multi-Link Subnet

Figure 7 illustrates the repeating IPv6 signaling that enables a 6LN to keep a global address alive and registered to its 6LBR using 6LoWPAN ND to the 6LR, RPL to the RPL Root, and then 6LoWPAN ND again to the 6BBR, which avoids repeating the Extended DAR/DAC flow across the network when RPL can suffice as a keep-alive mechanism.

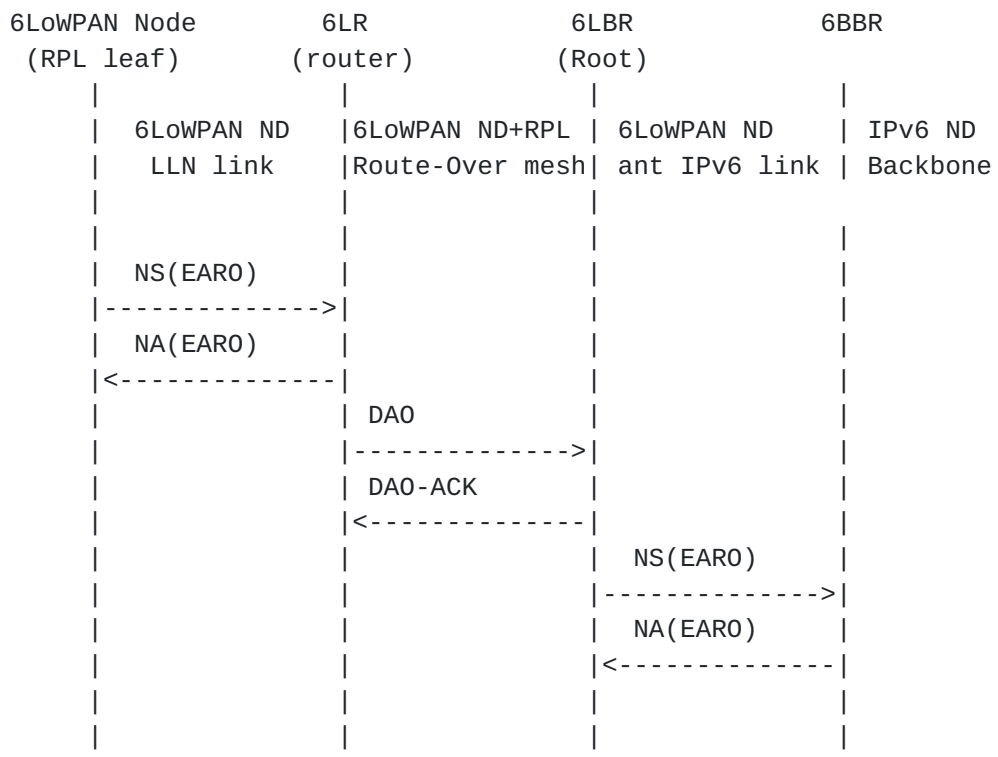

Figure 7: Next Registration Flow over Multi-Link Subnet

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.

4.3. TSCH and 6top

4.3.1. 6top

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 the spectrum in its vicinity.

With 6TiSCH, the abstraction of an IPv6 link is implemented as a pair of bundles of cells, one in each direction. IP Links are only enabled between RPL parents and children. The 6TiSCH operation is optimal when the size of a bundle is such that both the energy wasted in idle listening and the packet drops due to congestion loss are minimized, while packets are forwarded within an acceptable latency.

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. The 6TiSCH operation can remain optimal if RPL parents can adjust dynamically, and with enough reactivity to match the variations of best-effort traffic, the amount of bandwidth that is used to communicate between themselves and their children, in both directions. 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.

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 protocol, 6P, which is specified in [RFC8480], is one of the services provided by 6top. In particular, the 6top services are available over a management API that enables an external management entity to schedule cells and slotframes, and allows the addition of complementary functionality, for instance a Scheduling Function that manages a dynamic schedule management based on observed resource usage as discussed in Section 4.4.2. For this purpose, the 6TiSCH architecture differentiates "soft" cells and "hard" cells.

4.3.1.1. Hard Cells

"Hard" cells are cells that are owned and managed by a 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.

4.3.1.2. Soft Cells

In contrast, "soft" cells are cells that 6top can manage locally. 6top contains a monitoring process which monitors the performance of cells, and can add, remove soft cells in the TSCH schedule to adapt to the traffic needs, or move one when it performs poorly. 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.

4.3.2. Scheduling Functions and the 6top 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.

An 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 using the 6top protocol as illustrated in Figure 8.

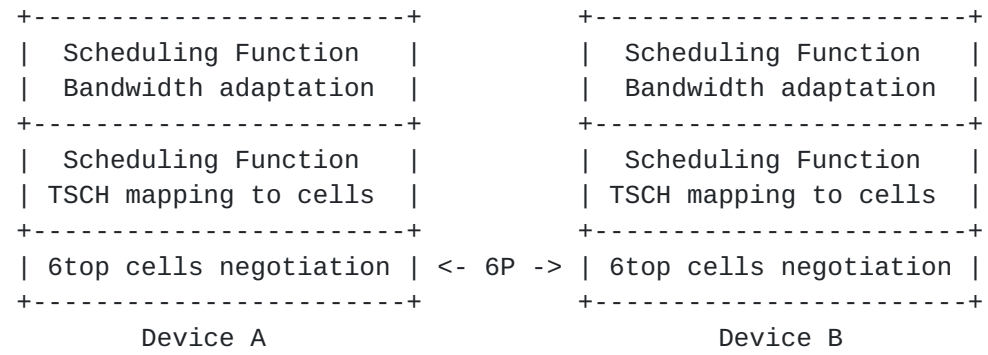

Figure 8: SF/6P stack in 6top

The SF relies on 6top services that implement the 6top Protocol (6P) [RFC8480] 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. 6P enables the peers to find an agreement in a transactional manner that ensures the final consistency of the nodes state.

MSF [I-D.ietf-6tisch-msf] is one of the possible scheduling functions. MSF uses the rendez-vous slot from [RFC8180] for network discovery, neighbor discovery, and any other broadcast.

For basic unicast communication with any neighbor, each node uses a receive cell at a well-known slotOffset/channelOffset, derived from a hash of their own MAC address. Nodes can reach any neighbor by installing a transmit (shared) cell with slotOffset/channelOffset derived from the neighbor's MAC address.

For child-parent links, MSF continuously monitors the load to/from parents and children. It then uses 6P to install/remove unicast cells whenever the current schedule appears to be under-/over-provisioned.

4.3.3. 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 [RFC8180] to support RPL over a static schedule, may leverage, for its internal computation, the information maintained by 6top.

An OF may require metrics about reachability, such as the Expected Transmission Count (ETX) metric [RFC6551]. 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. A neighbor table entry may contain a set of statistics with respect to that specific neighbor.

The neighbor information may include the time when the last packet has been received from that neighbor, a set of cell quality metrics, e.g., received signal strength indication (RSSI) or link quality indicator (LQI), the number of packets sent to the neighbor or the number of packets received from it. This information can be made available through 6top management APIs 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 behavior, for instance by configuring the periodicity of IEEE Std. 802.15.4 Extended Beacons (EBs). 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 Layer-2 Enhanced Beacons (EBs), which would produce undue timer coupling among layers, packet size issues and could conflict with the policy of production networks where EBs are mostly eliminated to conserve energy.

4.3.4. 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. 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-free structure. Nodes taken in a synchronization loop will rapidly desynchronize from the network and become isolated. 6TiSCH uses a RPL DAG with a dedicated global Instance 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".

The provisioning of a RPL Root is out of scope for both RPL and this Architecture, whereas RPL enables to propagate configuration information down the DODAG. This applies to the TSGI as well; a Root is configured or obtains by unspecified 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 in its IEEE Std. 802.15.4 Extended Beacons (EB).

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.

4.3.5. Slotframes and CDU matrix

6TiSCH enables IPv6 best effort (stochastic) transmissions over a MAC layer that is also capable of scheduled (deterministic) transmissions. A window of time is defined around the scheduled transmission where the medium must, as much as practically feasible, be free of contending energy to ensure that the medium is free of contending packets when time comes for a scheduled transmission. 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 IEEE Std. 802.15.4 TSCH as well as the Long Term Evolution (LTE) of cellular networks.

The 6TiSCH architecture defines a global concept that is called a Channel Distribution and Usage (CDU) matrix to describe that formatting of time and frequencies,

A CDU matrix is defined centrally as part of the network definition. It is a matrix of cells with a 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. There are different models for scheduling the usage of the cells, which place the responsibility of avoiding collisions either on a central controller or on the devices themselves, at an extra cost in terms of energy to scan for free cells (more in Section 4.4).

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 well-known channel rotation called the hopping sequence. 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 the 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. A slotframe is associated with a priority that may be related to the precedence of different 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.

4.3.6. Distributing the reservation of cells

The 6TiSCH architecture introduces the concept of chunks (Section 2.1) to distribute the allocation of the spectrum for a whole group of cells at a time. The CDU matrix is formatted into a set of chunks, possibly as illustrated in Figure 9, each of the chunks identified uniquely by a chunk-ID. The knowledge of this formatting is shared between all the nodes in a 6TiSCH network. It could be conveyed during the join process, or codified into a profile document, or obtained using some other mechanism. This is as opposed to static scheduling that refers to the pre-programmed mechanism that is specified in [RFC8180] and pre-exists to the distribution of the chunk formatting.

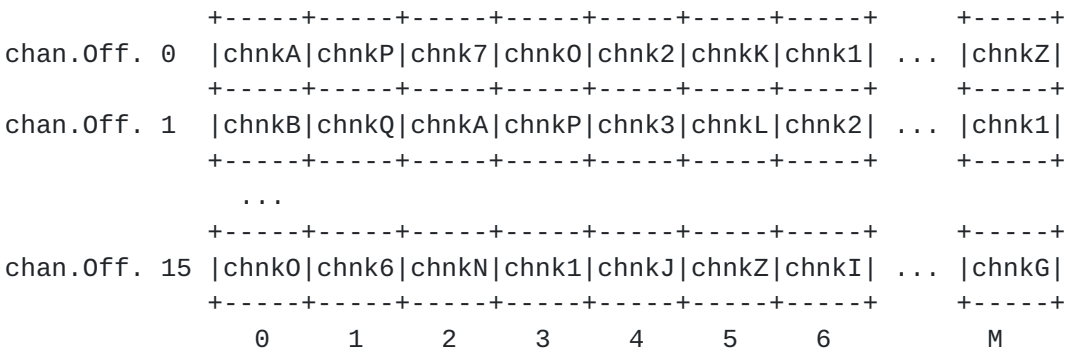

Figure 9: CDU matrix Partitioning in Chunks

The 6TiSCH Architecture envisions a protocol that enables chunk ownership appropriation whereby a RPL parent discovers a chunk that

is not used in its interference domain, 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.

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.

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.

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.

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 IEEE Std. 802.15.4 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 [RFC8180].

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.

As discussed in more details in Section 4.3, the 6top Protocol [RFC8480] specifies the exchanges between neighbor nodes to reserve soft cells to transmit to one another, possibly under the control of a Scheduling Function (SF). Because this reservation is done without global knowledge of the schedule of other nodes in the LLN, scheduling collisions are possible.

And as discussed in Section 4.3.2, an optional Scheduling Function (SF) 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. The "6TiSCH Minimal Scheduling Function (MSF)" [I-D.ietf-6tisch-msf] provides a simple scheduling function that can be used by default by devices that support dynamic scheduling of soft cells.

Monitoring and relocation is done in the 6top layer. For the upper layer, the connection between two neighbor nodes appears as a 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

Remote monitoring and Schedule Management refers to a DetNet/SDN model whereby an NME and a scheduling entity, associated with a PCE, reside in a central controller and interact with the 6top layer to control IPv6 Links and Tracks (Section 4.5) in a 6TiSCH network. The composite centralized controller can assign physical resources (e.g., buffers and hard cells) to a particular Track to optimize the reliability within a bounded latency for a well-specified flow.

The work at the 6TiSCH WG focused on non-deterministic traffic and did not provide the generic data model that is necessary for the controller to monitor and manage resources of the 6top sublayer. This is deferred to future work, see Appendix A.1.2.

With respect to Centralized routing and scheduling, it is envisioned that the related component of the 6TiSCH Architecture would be an extension of the Deterministic Networking Architecture [I-D.ietf-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 Software Defined Networking (SDN) Architecture and is composed of three planes, a (User) Application Plane, a Controller Plane (where the PCE operates), and a Network Plane which can represent a 6TiSCH LLN.

Software-Defined Networking (SDN): Layers and Architecture Terminology [RFC7426] proposes a generic representation of the SDN architecture that is reproduced in Figure 10.

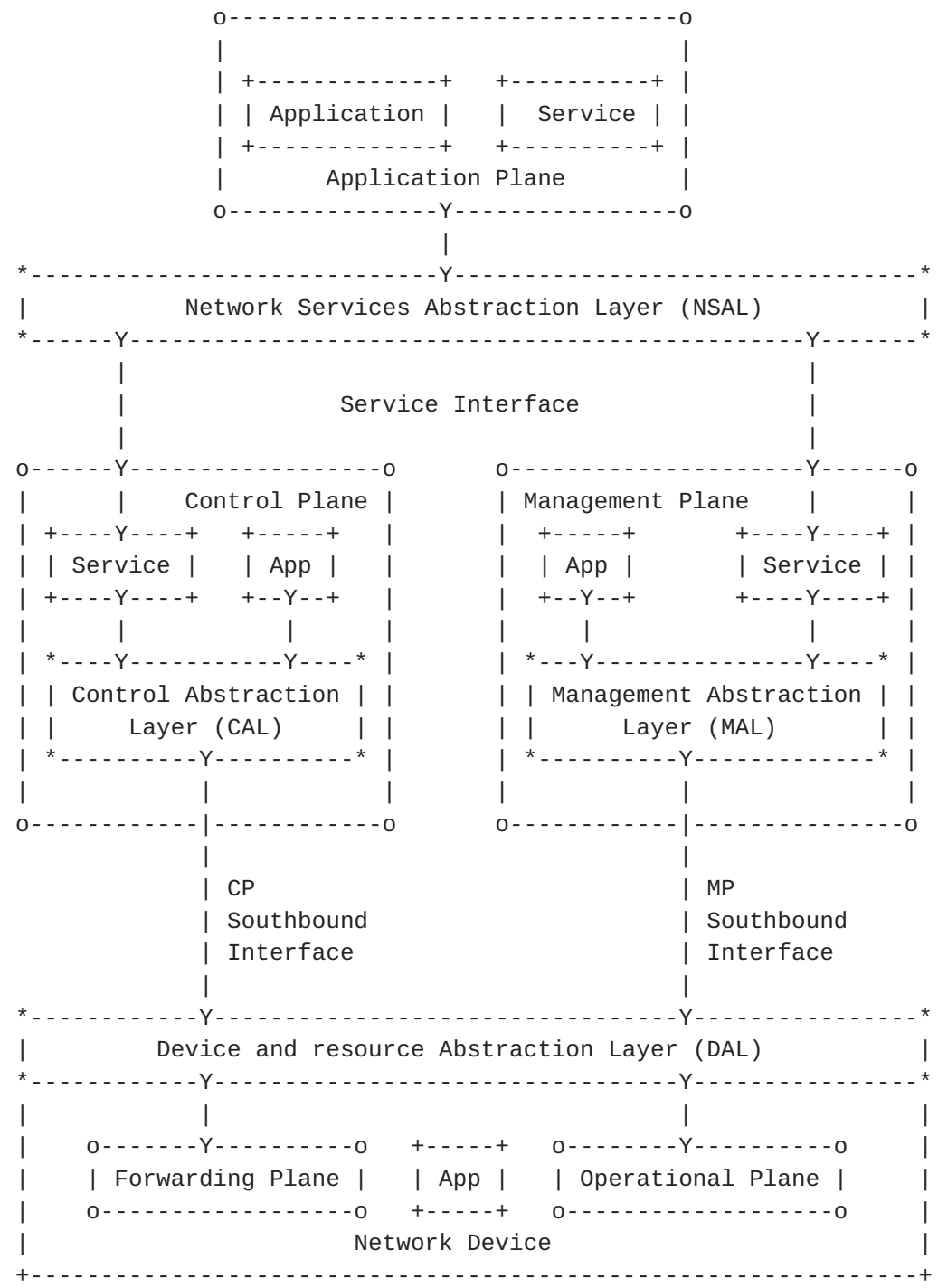

Figure 10: SDN Layers and Architecture Terminology per RFC 7426

The PCE establishes end-to-end Tracks of hard cells, which are described in more details in Section 4.6.1.

The DetNet work is expected to enable end to end Deterministic Path across heterogeneous network. This can be for instance 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 IEEE Std. 802.1 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 (Section 4.5) to one or more destination(s) that are multiple hops away by installing soft cells at each intermediate node. This forms a Track of soft cells. A Track Scheduling Function above the 6top sublayer of each node on the Track is needed 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. On Tracks

The architecture introduces the concept of a Track, which is a directed path from a source 6TiSCH node to one or more destination 6TiSCH node(s) across a 6TiSCH LLN.

A Track is the 6TiSCH instantiation of the concept of a Deterministic Path as described in [I-D.ietf-detnet-architecture]. Constrained resources such as memory buffers are reserved for that Track in intermediate 6TiSCH nodes to avoid loss related to limited capacity. A 6TiSCH node along a Track not only knows which bundles of cells it should use to receive packets from a previous hop, but also knows which bundle(s) it should use to send packets to its next hop along the Track.

4.5.1. General Behavior of Tracks

A Track is associated with Layer-2 bundles of cells with related schedules and logical relationships and that ensure that a packet that is injected in a Track will progress in due time all the way to destination.

Multiple cells may be scheduled in a Track for the transmission of a single packet, in which case the normal operation of IEEE Std. 802.15.4 Automatic Repeat-reQuest (ARQ) can take place; the acknowledgment may be omitted in some cases, for instance if there is no scheduled cell for a possible retry.

There are several benefits for using a Track to forward a packet from a source node to the destination node.

1. Track forwarding, as further described in Section 4.6.1, is a Layer-2 forwarding scheme, which introduces less process delay and overhead than Layer-3 forwarding scheme. Therefore, LLN Devices can save more energy and resource, which is critical for resource constrained devices.

2. Since channel resources, i.e., bundles of cells, have been reserved for communications between 6TiSCH nodes of each hop on the Track, the throughput and the maximum latency of the traffic along a Track are guaranteed and the jitter is maintained small.

3. By knowing the scheduled time slots of incoming bundle(s) and outgoing bundle(s), 6TiSCH nodes on a Track could save more energy by staying in sleep state during in-active slots.

4. Tracks are protected from interfering with one another if a cell is scheduled to belong to at most one Track, and congestion loss is avoided if at most one packet can be presented to the MAC to use that cell. Tracks enhance the reliability of transmissions and thus further improve the energy consumption in LLN Devices by reducing the chances of retransmission.

4.5.2. Serial Track

A Serial (or simple) Track is the 6TiSCH version of a circuit; a bundle of cells that are programmed 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 which can be used regardless of the network layer protocol. A Serial 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.

For a given iteration of the device schedule, the effective channel of the cell is obtained by following in a loop a well-known hopping sequence that started at Epoch time at 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 in the iteration of the schedule.

4.5.3. Complex Track with Replication and Elimination

The art of Deterministic Networks already include Packet Replication and Elimination techniques. Example standards include the Parallel Redundancy Protocol (PRP) and the High-availability Seamless Redundancy (HSR) [IEC62439]. Similarly, and as opposed to a Serial Track that is a sequence of nodes and links, a Complex Track is shaped as a directed acyclic graph towards one or more destination(s) to support multi-path forwarding and route around failures.

A Complex Track may branch off over non congruent branches for the purpose of multicasting, and/or redundancy, in which case it reconverges later down the path. This enables the Packet Replication, Elimination and Ordering Functions (PREOF) defined by Detnet. Packet ARQ, Replication, Elimination and Overhearing (PAREO) adds radio-specific capabilities of Layer-2 ARQ and promiscuous listening to redundant transmissions to compensate for the lossiness of the medium and meet industrial expectations of a Reliable and Available Wireless network. Combining PAREO and PREOF, a Track may extend beyond the 6TiSCH network in a larger DetNet network.

In the art of TSCH, a path does not necessarily support PRE but it is almost systematically multi-path. This means that a Track is scheduled so as to ensure that each hop has at least two forwarding solutions, and the forwarding decision is to try the preferred one and use the other in case of Layer-2 transmission failure as detected by ARQ. Similarly, at each 6TiSCH hop along the Track, the PCE may schedule more than one timeslot for a packet, so as to support Layer-2 retries (ARQ). It is also possible that the field device only uses the second branch if sending over the first branch fails.

4.5.4. DetNet End-to-end Path

Ultimately, DetNet [I-D.ietf-detnet-architecture] should enable to extend a Track beyond the 6TiSCH LLN as illustrated in Figure 11. In that example, a Track that is laid out from a field device in a 6TiSCH network to an IoT gateway that is located on an 802.1 Time-Sensitive Networking (TSN) backbone. A 6TiSCH-Aware DetNet Service Layer handles the Packet Replication, Elimination, and Ordering Functions over the DODAG that forms a Track.

The Replication function in the 6TiSCH Node sends a copy of each packet over two different branches, and the PCE schedules each hop of both branches so that the two copies arrive in due time at the gateway. In case of a loss on one branch, hopefully the other copy of the packet still makes it in due time. If two copies make it to the IoT gateway, the Elimination function in the gateway ignores the extra packet and presents only one copy to upper layers.

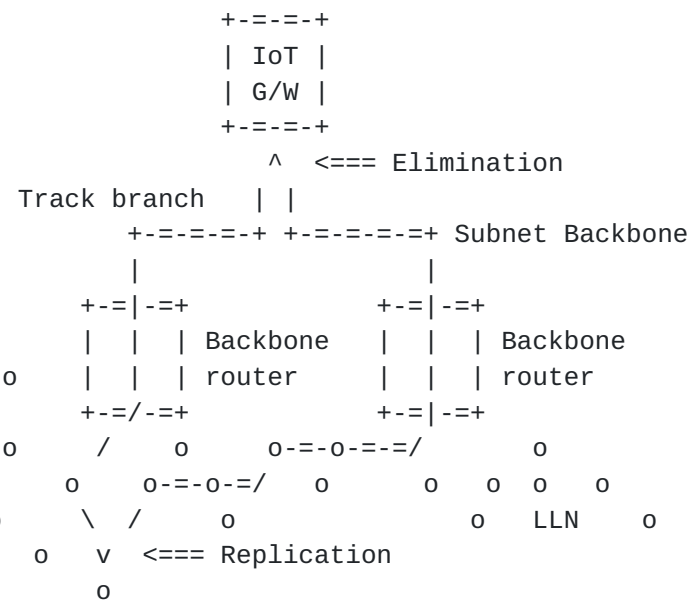

Figure 11: Example End-to-End DetNet Track

4.5.5. Cell Reuse

The 6TiSCH architecture provides means to avoid waste of cells as well as overflows in the transmit bundle of a Track, as follows:

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 in a frame that is received in a RX-cell of a Track with a destination MAC address set to this node as opposed to the broadcast MAC address must be extracted from the Track and delivered to the upper layer. Note that a frame with an unrecognized destination 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, and if the frame transports an IPv6 packet, then it can be placed for transmission in the bundle that is used for Layer-3 traffic towards the next hop along the Track. The MAC address should be set to the next-hop MAC address to avoid confusion.

It results in 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 is re-Tracked by scheduling it for transmission over the transmit bundle associated to the Track, with the destination MAC address set to broadcast.

4.6. Forwarding Models

By forwarding, this document 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 4.7. 6TiSCH supports three different forwarding model:(G-MPLS) Track Forwarding, (classical) IPv6 Forwarding and (6LoWPAN) Fragment Forwarding.

4.6.1. Track Forwarding

Forwarding along a Track can 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 IEEE Std. 802.15.4, this means effectively broadcast, so that along the Track the short address for the destination of the frame is set to 0xFFFF.

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

4.6.1.1. Native Mode

In native mode, the Protocol Data Unit (PDU) is associated with flow-dependent 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 may be done using a 6-tuple as discussed in [I-D.ietf-detnet-ip]. In particular, implementations of this document should support identification of DetNet flows based on the IPv6 Flow Label field. The flow identification may also be done using a dedicated RPL Instance (see section 3.1.3 of [RFC6550]), signaled in a RPL Packet Information (more in section 11.2.2.1 of [RFC6550]). The flow identification is validated at egress before restoring the destination MAC address (DMAC) and punting to the upper layer.

Figure 12 illustrates the Track Forwarding operation which happens at the 6top sublayer, below IP.

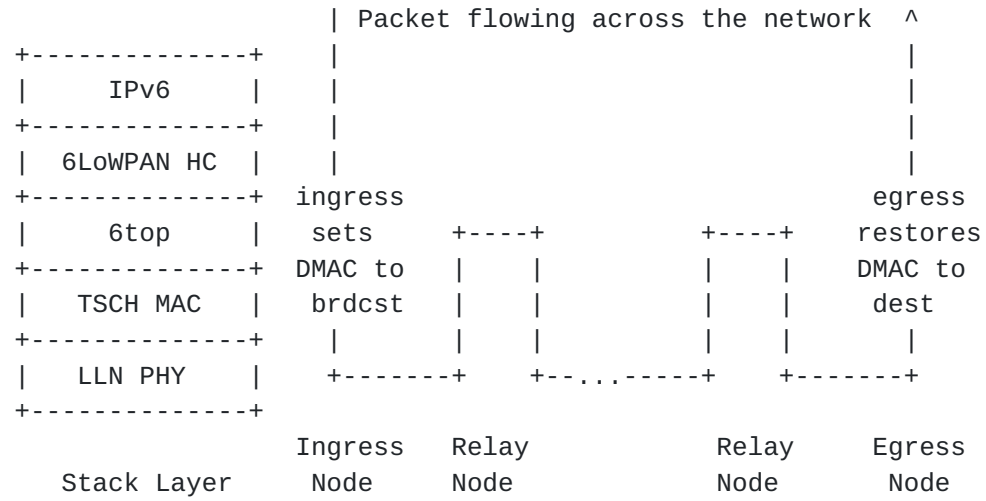

Figure 12: Track Forwarding, Native Mode

4.6.1.2. Tunnel Mode

In tunnel mode, the frames originate from an arbitrary protocol over a compatible MAC that may or may not be 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 transported.

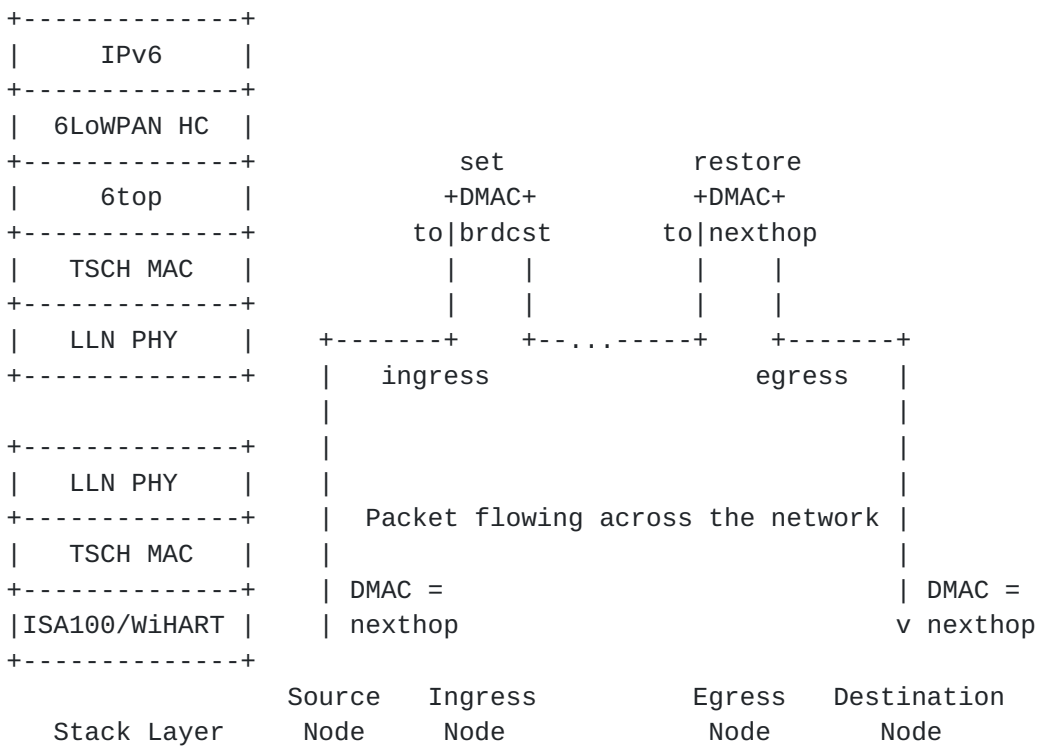

Figure 13: Track Forwarding, Tunnel Mode

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 tunneling information of the Track, which may for instance indicate that the tunneled datagram is an IP packet, the datagram is passed to the appropriate Link-Layer with the destination MAC restored.

4.6.1.3. Tunneling Information

Tunneling information coming with the Track configuration provides 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.

If the Layer-3 destination address belongs to 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. Restoring the wrong MAC address at the egress would then also result in the wrong IP address in the packet after decompression. For that reason, a packet can be injected in a Track only if the destination MAC address is effectively that of the tunnel egress point. It is thus mandatory for the ingress router to validate that the MAC address that was used at the 6LoWPAN sublayer for compression matches 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 information.

4.6.2. IPv6 Forwarding

As the packets are routed at Layer-3, traditional QoS and Active Queue Management (AQM) operations are expected to prioritize flows.

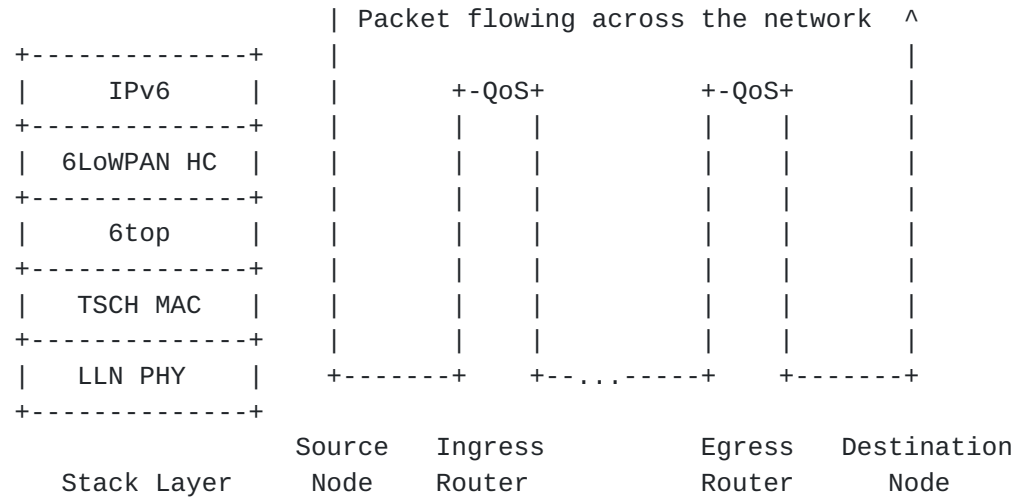

Figure 14: IP Forwarding

4.6.3. Fragment Forwarding

Considering that per section 4 of [RFC4944] 6LoWPAN packets can be as large as 1280 bytes (the IPv6 minimum MTU), and that the non-storing mode of RPL implies Source Routing that requires space for routing headers, and that a IEEE Std. 802.15.4 frame with security may carry in the order of 80 bytes of effective payload, an IPv6 packet might be fragmented into more than 16 fragments at the 6LoWPAN sublayer.

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

In the case to a multihop route within a 6TiSCH network, Hop-by-Hop recomposition occurs at each hop 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.ietf-6lo-minimal-fragment] describes a framework for forwarding fragments end-to-end across a 6TiSCH route-over mesh. Within that framework, [I-D.ietf-lwig-6lowpan-virtual-reassembly] details a virtual reassembly buffer mechanism whereby the datagram tag in the 6LoWPAN Fragment is used as a label for switching at the 6LoWPAN sublayer.

Building on this technique, [I-D.ietf-6lo-fragment-recovery] introduces a new format for 6LoWPAN fragments that enables the selective recovery of individual fragments, and allows for a degree of flow control based on an Explicit Congestion Notification.

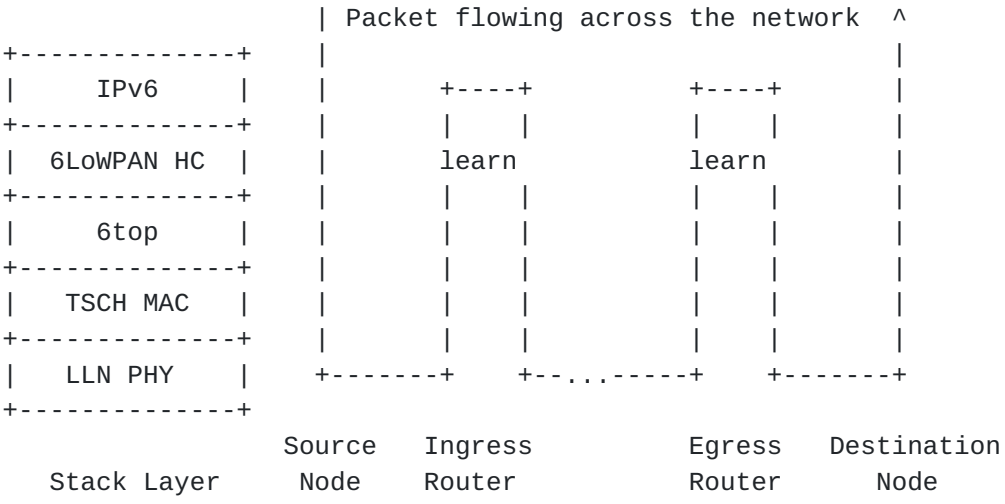

Figure 15: Forwarding First Fragment

In that model, the first fragment is routed based on the IPv6 header that is present in that fragment. The 6LoWPAN sublayer learns the next hop selection, generates a new datagram tag for transmission to the next hop, and stores that information indexed by the incoming MAC address and datagram tag. The next fragments are then switched based on that stored state.

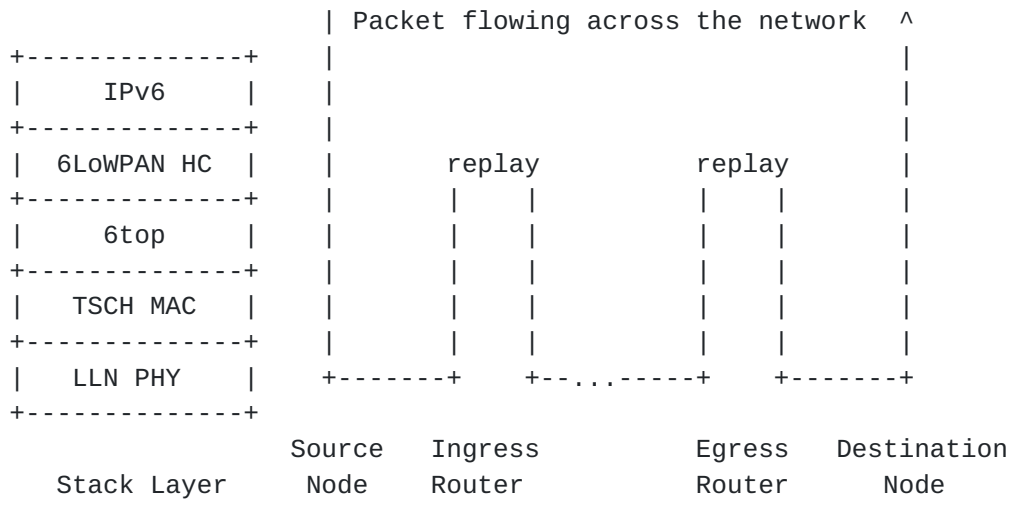

Figure 16: Forwarding Next Fragment

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.

4.7. Advanced 6TiSCH Routing

4.7.1. Packet Marking and Handling

All packets inside a 6TiSCH domain must carry the RPLInstanceID 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.

For packets that are routed by RPL, that information is the RPLInstanceID which is carried in the RPL Packet Information (RPI), as discussed in section 11.2 of [RFC6550], "Loop Avoidance and Detection". The RPI is transported by 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 specified in [RFC8025] and [RFC8138].

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.

4.7.2. Replication, Retries and Elimination

6TiSCH supports the PREOF operations of elimination and reordering of packets along a complex Track, but has no requirement about whether a sequence number is tagged in the packet for that purpose. With 6TiSCH, the schedule can tell when multiple receive timeslots correspond to copies of a same packet, in which case the receiver may avoid listening to the extra copies once it had received one instance of the packet.

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

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

On the receive side, all timeslots are programmed in a same 'OR' group. Retries of a same copy as well as converging branches for elimination are converged, meaning that the first successful reception is enough and that all the other timeslots can be ignored. A 'AND' group denotes different packets that must all be received and transmitted over the associated transmit groups within their respected 'AND' or 'OR' rules.

As an example say that we have a simple network as represented in Figure 17, and we want to enable PREOF between an ingress node I and an egress node E.

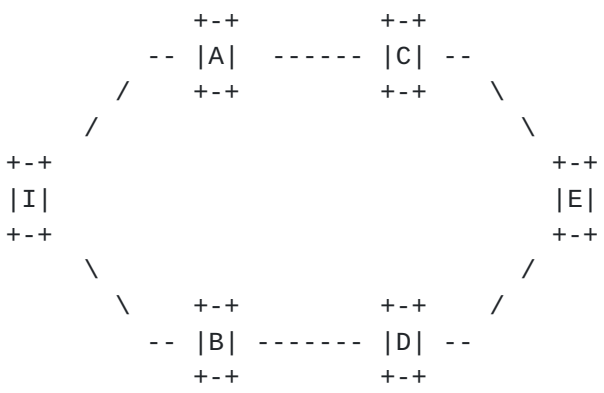

Figure 17: Scheduling PREOF on a Simple Network

The assumption for this particular problem is that a 6TiSCH node has a single radio, so it cannot perform 2 receive and/or transmit operations at the same time, even on 2 different channels.

Say we have 6 possible channels, and at least 10 timeslots per slotframe. Figure 18 shows a possible schedule whereby each transmission is retried 2 or 3 times, and redundant copies are forwarded in parallel via A and C on the one hand, and B and D on the other, providing time diversity, spatial diversity though different physical paths, and frequency diversity.

slotOffset	0	1	2	3	4	5	6	7	9	
channelOffset 0							B->D			...
channelOffset 1		I->A		A->C	B->D					...
channelOffset 2	I->A			I->B		C->E		D->E		...
channelOffset 3					A->C					...
channelOffset 4			I->B			B->D			D->E	...
channelOffset 5			A->C				C->E			...

Figure 18: Example Global Schedule

This translates in a different slotframe for every node that provides the waking and sleeping times, and the channelOffset to be used when awake. Figure 19 shows the corresponding slotframe for node A.

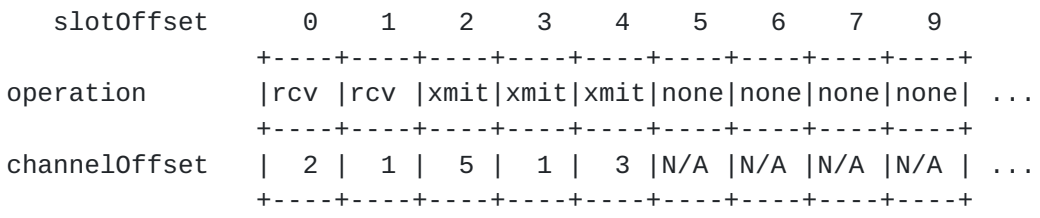

Figure 19: Example Slotframe for Node A

The logical relationship between the timeslots is given by the following table:

Node	rcv slotOffset	xmit slotOffset
I	N/A	(0 OR 1) AND (2 OR 3)
A	(0 OR 1)	(2 OR 3 OR 4)
B	(2 OR 3)	(4 OR 5 OR 6)
C	(2 OR 3 OR 4)	(5 OR 6)
D	(4 OR 5 OR 6)	(7 OR 8)
E	(5 OR 6 OR 7 OR 8)	N/A

5. IANA Considerations

This document does not require IANA action.

6. Security Considerations

The "Minimal Security Framework for 6TiSCH" [I-D.ietf-6tisch-minimal-security] was optimized for Low-Power and TSCH operations. The reader is encouraged to review the Security Considerations section of that document, which discusses 6TiSCH security issues in more details.

6.1. Availability of Remote Services

The operation of 6TiSCH Tracks inherits its high level operation from DetNet and is subject to the observations in section 5 of [I-D.ietf-detnet-architecture]. The installation and the maintenance

of the 6TiSCH Tracks depends on the availability of a controller with a PCE to compute and push them in the network. When that connectivity is lost, existing Tracks may continue to operate until the end of their lifetime, but cannot be removed or updated, and new Tracks cannot be installed.

In a LLN, the communication with a remote PCE may be slow and unreactive to rapid changes in the condition of the wireless communication. An attacker may introduce extra delay by selectively jamming some packets or some flows. The expectation is that the 6TiSCH Tracks enable enough redundancy to maintain the critical traffic in operation while new routes are calculated and programmed into the network.

As with DetNet in general, the communication with the PCE must be secured and should be protected against DoS attacks, including delay injection and blackholing attacks, and secured as discussed in the security considerations defined for Abstraction and Control of Traffic Engineered Networks (ACTN) in Section 9 of [RFC8453], which applies equally to DetNet and 6TiSCH. In a similar manner, the communication with the JRC must be secured and should be protected against DoS attacks when possible.

6.2. Selective Jamming

The Hopping Sequence of a TSCH network is well-known, meaning that if a rogue manages to identify a cell of a particular flow, then it may to selectively jam that cell, without impacting any other traffic. This attack can be performed at the PHY layer without any knowledge of the Layer-2 keys, and is very hard to detect and diagnose because only one flow is impacted.

[I-D.tiloca-6tisch-robust-scheduling] proposes a method to obfuscate the hopping sequence and make it harder to perpetrate that particular attack.

6.3. MAC-Layer Security

This architecture operates on IEEE Std. 802.15.4 and expects the 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. In a typical deployment, all joined nodes use the same keys and rekeying needs to be global.

The 6TISCH Architecture relies on the join process to deny authorization of invalid nodes and preserve the integrity of the network keys. A rogue that managed to access the network can perform

a large variety of attacks from DoS to injecting forged packets and routing information. "Zero-trust" properties would be highly desirable but are mostly not available at the time of this writing. [I-D.ietf-6lo-ap-nd] is a notable exception that protects the ownership of IPv6 addresses and prevents a rogue node with L2 access from stealing and injecting traffic on behalf of a legitimate node.

6.4. Time Synchronization

Time Synchronization in TSCH induces another event horizon whereby a node will only communicate with another node if they are synchronized within a guard time. The pledge discovers the synchronization of the network based on the time of reception of the beacon. If an attacker synchronizes a pledge outside of the guard time of the legitimate nodes then the pledge will never see a legitimate beacon and may not discover the attack.

As discussed in [I-D.ietf-detnet-architecture], measures must be taken to protect the time synchronization, and for 6TiSCH this includes ensuring that the Absolute Slot Number (ASN), which is the node's sense of time, is not compromised. Once installed and as long as the node is synchronized to the network, ASN is implicit in the transmissions.

IEEE Std. 802.15.4 [IEEE802154] specifies that in a TSCH network, the nonce that is used for the computation of the Message Integrity Code (MIC) to secure Link-Layer frames is composed of the address of the source of the frame and of the ASN. The standard assumes that the ASN is distributed securely by other means. The ASN is not passed explicitly in the data frames and does not constitute a complete anti-replay protection. It results that upper layer protocols must provide a way to detect duplicates and cope with them.

If the receiver and the sender have a different sense of ASN, the MIC will not validate and the frame will be dropped. In that sense, TSCH induces an event horizon whereby only nodes that have a common sense of ASN can talk to one another in an authenticated manner. With 6TiSCH, the pledge discovers a tentative ASN in beacons from nodes that have already joined the network. But even if the beacon can be authenticated, the ASN cannot be trusted as it could be a replay by an attacker and thus could announce an ASN that represents a time in the past. If the pledge uses an ASN that is learned from a replayed beacon for an encrypted transmission, a nonce-reuse attack becomes possible and the network keys may be compromised.

6.5. Validating ASN

After obtaining the tentative ASN, a pledge that wishes to join the 6TiSCH network must use a join protocol to obtain its security keys. The join protocol used in 6TiSCH is the Constrained Join Protocol (CoJP). In the minimal setting defined in [I-D.ietf-6tisch-minimal-security], the authentication requires a pre-shared key, based on which a secure session is derived. The CoJP exchange may also be preceded with a zero-touch handshake [I-D.ietf-6tisch-dtsecurity-zerotouch-join] in order to enable pledge joining based on certificates and/or inter-domain communication.

As detailed in Section 4.2.1, a Join Proxy (JP) helps the pledge for the join procedure by relaying the link-scope Join Request over the IP network to a Join Registrar/Coordinator (JRC) that can authenticate the pledge and validate that it is attached to the appropriate network. As a result of the CoJP exchange, the pledge is in possession of a Link-Layer material including keys and a short address, and if the ASN is known to be correct, all traffic can now be secured using CCM* [CCMstar] at the Link-Layer.

The authentication steps must be such that they cannot be replayed by an attacker, and they must not depend on the tentative ASN being valid. During the authentication, the keying material that the pledge obtains from the JRC does not provide protection against spoofed ASN. Once the pledge has obtained the keys to use in the network, it may still need to verify the ASN. If the nonce used in the Layer-2 security derives from the extended (MAC-64) address, then replaying the ASN alone cannot enable a nonce-reuse attack unless the same node is lost its state with a previous ASN. But if the nonce derives from the short address (e.g., assigned by the JRC) then the JRC must ensure that it never assigns short addresses that were already given to this or other nodes with the same keys. In other words, the network must be rekeyed before the JRC runs out of short addresses.

6.6. Network Keying and Rekeying

Section 4.2.1 provides an overview of the CoJP process described in [I-D.ietf-6tisch-minimal-security] by which an LLN can be assembled in the field, having been provisioned in a lab. [I-D.ietf-6tisch-dtsecurity-zerotouch-join] is future work that preceeds and then leverages the CoJP protocol using the [I-D.ietf-anima-constrained-voucher] constrained profile of [I-D.ietf-anima-bootstrapping-keyinfra] (BRSKI). This later work requires a yet-to-be standardized Lighweight Authenticated Key Exchange protocol.

The CoJP protocol results in distribution of a network-wide key that is to be used with [IEEE802154] security. The details of use are described in [I-D.ietf-6tisch-minimal-security] sections 9.2 and 9.3.2.

The BRSKI mechanism may lead to the use of the CoJP protocol, in which case it also results in distribution of a network-wide key. Alternatively the BRSKI mechanism may be followed by use of [I-D.ietf-ace-coap-est] to enroll certificates for each device. In that case, the certificates may be used with an [IEEE802154] key agreement protocol. The description of this mechanism, while conceptually straight forward still has significant standardization hurdles to pass.

[I-D.ietf-6tisch-minimal-security] section 9.2 describes a mechanism to change (rekey) the network. There are a number of reasons to initiate a network rekey: to remove unwanted (corrupt/malicious) nodes, to recover unused 2-byte short addresses, or due to limits in encryption algorithms. For all of the mechanisms that distribute a network-wide key, rekeying is also needed on a periodic basis. In more details:

- The mechanism described in [I-D.ietf-6tisch-minimal-security] section 9.2 requires advance communication between the JRC and every one of the nodes before the key change. Given that many nodes may be sleepy, this operation may take a significant amount of time, and may consume a significant portion of the available bandwidth. As such, network-wide rekeys in order to exclude nodes that have become malicious will not be particularly quick. If a rekey is already in progress, but the unwanted node has not yet been updated, then it is possible to to just continue the operation. If the unwanted node has already received the update, then the rekey operation will need to be restarted.

- The cryptographic mechanisms used by IEEE Std. 802.15.4 include the 2-byte short address in the calculation of the context. A nonce-reuse attack may become feasible if a short address is reassigned to another node while the same network-wide keys are in operation. A network that gains and loses nodes on a regular basis is likely to reach the 65536 limit of the 2-byte (16-bit) short addresses, even if the network has only a few thousand nodes. Network planners should consider the need to rekey the network on a periodic basis in order to recover 2-byte addresses. The rekey can update the short addresses for active nodes if desired, but there is actually no need to do this as long as the key has been changed.

- With TSCH as it stands at the time of this writing, the ASN will wrap after 2^40 timeslot durations, which means with the default values around 350 years. Wrapping ASN is not expected to happen within the lifetime of most LLNs. Yet, should the ASN wrap, the network must be rekeyed to avoid a nonce-reuse attack.

- Many cipher algorithms have some suggested limits on how many bytes should be encrypted with that algorithm before a new key is used. These numbers are typically in the many to hundreds of gigabytes of data. On very fast backbone networks this becomes an important concern. On LLNs with typical data rates in the kilobits/second, this concern is significantly less. With IEEE Std. 802.15.4 as it stands at the time of this writing, the ASN will wrap before the limits of the current L2 crypto (AES-CCM-128) are reached, so the problem should never occur.

- In any fashion, if the LLN is expected to operate continuously for decades then the operators are advised to plan for the need to rekey.

Except for urgent rekeys caused by malicious nodes, the rekey operation described in [I-D.ietf-6tisch-minimal-security] can be done as a background task and can be done incrementally. It is a make-before-break mechanism. The switch over to the new key is not signaled by time, but rather by observation that the new key is in use. As such, the update can take as long as needed, or occur in as short a time as practical.

7. Acknowledgments

7.1. Contributors

The co-authors of this document are listed below:

Thomas Watteyne for his contribution to the whole design, in particular on TSCH and security, and to the open source community with openWSN that he created.

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;

Kris Pister for creating TSCH and his continuing guidance through the elaboration of this design;

Malisa Vucinic for the work on the one-touch join process and his contribution to the Security Design Team;

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

Tero Kivinen for his contribution to the security work in general and the security section in particular.

Maria Rita Palattella for managing the Terminology document merged into this through the work of 6TiSCH;

Simon Duquennoy for his contribution to the open source community with the 6TiSCH implementaton of contiki, and for his contribution to MSF and autonomous unicast cells.

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

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

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

7.2. Special Thanks

Special thanks to Jonathan Simon, Giuseppe Piro, Subir Das and Yoshihiro Ohba for their deep contribution to the initial security work, to Yasuyuki Tanaka for his work on implementation and simulation that tremendously helped build a robust system, to Diego Dujovne for starting and leading the SF0 effort and to Tengfei Chang for evolving it in the MSF.

Special thanks also to Pat Kinney, Charlie Perkins and Bob Heile for their support in maintaining the connection active and the design in line with work happening at IEEE 802.15.

Special thanks to Ted Lemon who was the INT Area A-D while this document was initiated for his great support and help throughout, and to Suresh Krishnan who took over with that kind efficiency of his till publication.

Also special thanks to Ralph Droms who performed the first INT Area Directorate review, that was very deep and thorough and radically changed the orientations of this document, and then to Eliot Lear and Carlos Pignataro who help finalize this document in preparation to the IESG reviews, and to Gorry Fairhurst, David Mandelberg, Qin Wu, Francis Dupont, Eric Vyncke, Mirja Kuhlewind, Roman Danyliw, Benjamin Kaduk and Andrew Malis, who contributed to the final shaping of this document through the IESG review procedure.

7.3. And Do not Forget

This document is the result of multiple interactions, in particular during the 6TiSCH (bi)Weekly Interim call, relayed through the 6TiSCH mailing list at the IETF, over the course of more than 5 years.

The authors wish to thank in arbitrary order: Alaeddine Weslati, Chonggang Wang, Georgios Exarchakos, Zhuo Chen, Georgios Papadopoulos, Eric Levy-Abegnoli, Alfredo Grieco, Bert Greevenbosch, Cedric Adjih, Deji Chen, Martin Turon, Dominique Barthel, Elvis Vogli, Geraldine Texier, Guillaume Gaillard, Herman Storey, Kazushi Muraoka, Ken Bannister, Kuor Hsin Chang, Laurent Toutain, Maik Seewald, 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, 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.ietf-6lo-ap-nd]
Thubert, P., Sarikaya, B., Sethi, M., and R. Struik, "Address Protected Neighbor Discovery for Low-power and Lossy Networks", draft-ietf-6lo-ap-nd-12 (work in progress), April 2019.

[I-D.ietf-6lo-backbone-router]
Thubert, P., Perkins, C., and E. Levy-Abegnoli, "IPv6 Backbone Router", draft-ietf-6lo-backbone-router-13 (work in progress), September 2019.

[I-D.ietf-6lo-fragment-recovery]
Thubert, P., "6LoWPAN Selective Fragment Recovery", draft-ietf-6lo-fragment-recovery-05 (work in progress), July 2019.

[I-D.ietf-6lo-minimal-fragment]
Watteyne, T., Bormann, C., and P. Thubert, "6LoWPAN Fragment Forwarding", draft-ietf-6lo-minimal-fragment-04 (work in progress), September 2019.

[I-D.ietf-6tisch-enrollment-enhanced-beacon]
Dujovne, D. and M. Richardson, "IEEE802.15.4 Informational Element encapsulation of 6tisch Join and Enrollment Information", draft-ietf-6tisch-enrollment-enhanced-beacon-05 (work in progress), September 2019.

[I-D.ietf-6tisch-minimal-security]
Vucinic, M., Simon, J., Pister, K., and M. Richardson, "Minimal Security Framework for 6TiSCH", draft-ietf-6tisch-minimal-security-12 (work in progress), July 2019.

[I-D.ietf-6tisch-msf]
Chang, T., Vucinic, M., Vilajosana, X., Duquennoy, S., and D. Dujovne, "6TiSCH Minimal Scheduling Function (MSF)", draft-ietf-6tisch-msf-06 (work in progress), August 2019.

[I-D.ietf-detnet-architecture]
Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-13 (work in progress), May 2019.

[I-D.ietf-roll-unaware-leaves]
Thubert, P. and M. Richardson, "Routing for RPL Leaves", draft-ietf-roll-unaware-leaves-04 (work in progress), September 2019.

[I-D.ietf-roll-useofrplinfo]
Robles, I., Richardson, M., and P. Thubert, "Using RPL Option Type, Routing Header for Source Routes and IPv6-in-IPv6 encapsulation in the RPL Data Plane", draft-ietf-roll-useofrplinfo-31 (work in progress), August 2019.

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

[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, <https://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, <https://www.rfc-editor.org/info/rfc4862>.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <https://www.rfc-editor.org/info/rfc4944>.

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

[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, <https://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, <https://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, <https://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, <https://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, <https://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, <https://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, <https://www.rfc-editor.org/info/rfc6775>.

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

[RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, DOI 10.17487/RFC7228, May 2014, <https://www.rfc-editor.org/info/rfc7228>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <https://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, <https://www.rfc-editor.org/info/rfc7554>.

[RFC8025] Thubert, P., Ed. and R. Cragie, "IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Paging Dispatch", RFC 8025, DOI 10.17487/RFC8025, November 2016, <https://www.rfc-editor.org/info/rfc8025>.

[RFC8137] Kivinen, T. and P. Kinney, "IEEE 802.15.4 Information Element for the IETF", RFC 8137, DOI 10.17487/RFC8137, May 2017, <https://www.rfc-editor.org/info/rfc8137>.

[RFC8138] Thubert, P., Ed., Bormann, C., Toutain, L., and R. Cragie, "IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing Header", RFC 8138, DOI 10.17487/RFC8138, April 2017, <https://www.rfc-editor.org/info/rfc8138>.

[RFC8180] Vilajosana, X., Ed., Pister, K., and T. Watteyne, "Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e (6TiSCH) Configuration", BCP 210, RFC 8180, DOI 10.17487/RFC8180, May 2017, <https://www.rfc-editor.org/info/rfc8180>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC 8453, DOI 10.17487/RFC8453, August 2018, <https://www.rfc-editor.org/info/rfc8453>.

[RFC8480] Wang, Q., Ed., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top) Protocol (6P)", RFC 8480, DOI 10.17487/RFC8480, November 2018, <https://www.rfc-editor.org/info/rfc8480>.

[RFC8505] Thubert, P., Ed., Nordmark, E., Chakrabarti, S., and C. Perkins, "Registration Extensions for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Neighbor Discovery", RFC 8505, DOI 10.17487/RFC8505, November 2018, <https://www.rfc-editor.org/info/rfc8505>.

8.2. Informative References

[AMI] US Department of Energy, "Advanced Metering Infrastructure and Customer Systems", 2006, <https://www.energy.gov/sites/prod/files/2016/12/f34/AMI%20Summary%20Report_09-26-16.pdf>.

[ANIMA] IETF, "Autonomic Networking Integrated Model and Approach", <https://dataTracker.ietf.org/doc/charter-ietf-anima/>.

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

[CCMstar] Struik, R., "Formal Specification of the CCM* Mode of Operation", September 2004, <www.ieee802.org/15/pub/2004/15-04-0537-00-004b-formal-specification-ccm-star-mode-operation.doc>.

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

[I-D.ietf-6tisch-dtsecurity-zerotouch-join] Richardson, M., "6tisch Zero-Touch Secure Join protocol", draft-ietf-6tisch-dtsecurity-zerotouch-join-04 (work in progress), July 2019.

[I-D.ietf-ace-coap-est]
Stok, P., Kampanakis, P., Richardson, M., and S. Raza, "EST over secure CoAP (EST-coaps)", draft-ietf-ace-coap-est-15 (work in progress), October 2019.

[I-D.ietf-anima-bootstrapping-keyinfra]
Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructures (BRSKI)", draft-ietf-anima-bootstrapping-keyinfra-28 (work in progress), September 2019.

[I-D.ietf-anima-constrained-voucher]
Richardson, M., Stok, P., and P. Kampanakis, "Constrained Voucher Artifacts for Bootstrapping Protocols", draft-ietf-anima-constrained-voucher-05 (work in progress), July 2019.

[I-D.ietf-core-object-security]
Selander, G., Mattsson, J., Palombini, F., and L. Seitz, "Object Security for Constrained RESTful Environments (OSCORE)", draft-ietf-core-object-security-16 (work in progress), March 2019.

[I-D.ietf-detnet-ip]
Varga, B., Farkas, J., Berger, L., Fedyk, D., Malis, A., Bryant, S., and J. Korhonen, "DetNet Data Plane: IP", draft-ietf-detnet-ip-01 (work in progress), July 2019.

[I-D.ietf-lwig-6lowpan-virtual-reassembly]
Bormann, C. and T. Watteyne, "Virtual reassembly buffers in 6LoWPAN", draft-ietf-lwig-6lowpan-virtual-reassembly-01 (work in progress), March 2019.

[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-aodv-rpl]
Anamalamudi, S., Zhang, M., Perkins, C., Anand, S., and B. Liu, "Asymmetric AODV-P2P-RPL in Low-Power and Lossy Networks (LLNs)", draft-ietf-roll-aodv-rpl-07 (work in progress), April 2019.

[I-D.ietf-roll-dao-projection]
Thubert, P., Jadhav, R., Gillmore, M., and J. Pylakutty, "Root initiated routing state in RPL", draft-ietf-roll-dao-projection-06 (work in progress), May 2019.

[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.pthubert-raw-problem-statement]
Thubert, P. and G. Papadopoulos, "Reliable and Available Wireless Problem Statement", draft-pthubert-raw-problem-statement-03 (work in progress), October 2019.

[I-D.rahul-roll-mop-ext]
Jadhav, R. and P. Thubert, "RPL Mode of Operation extension", draft-rahul-roll-mop-ext-01 (work in progress), June 2019.

[I-D.selander-ace-cose-ecdhe]
Selander, G., Mattsson, J., and F. Palombini, "Ephemeral Diffie-Hellman Over COSE (EDHOC)", draft-selander-ace-cose-ecdhe-14 (work in progress), September 2019.

[I-D.thubert-6lo-bier-dispatch]
Thubert, P., Brodard, Z., Jiang, H., and G. Texier, "A 6loRH for BitStrings", draft-thubert-6lo-bier-dispatch-06 (work in progress), January 2019.

[I-D.thubert-6man-unicast-lookup]
Thubert, P. and E. Levy-Abegnoli, "IPv6 Neighbor Discovery Unicast Lookup", draft-thubert-6man-unicast-lookup-00 (work in progress), July 2019.

[I-D.thubert-bier-replication-elimination]
Thubert, P., Eckert, T., Brodard, Z., and H. Jiang, "BIER-TE extensions for Packet Replication and Elimination Function (PREF) and OAM", draft-thubert-bier-replication-elimination-03 (work in progress), March 2018.

[I-D.thubert-raw-technologies]
Thubert, P., Cavalcanti, D., Vilajosana, X., and C. Schmitt, "Reliable and Available Wireless Technologies", draft-thubert-raw-technologies-03 (work in progress), July 2019.

[I-D.thubert-roll-bier]
Thubert, P., "RPL-BIER", draft-thubert-roll-bier-02 (work in progress), July 2018.

[I-D.tiloca-6tisch-robust-scheduling]
Tiloca, M., Duquennoy, S., and G. Dini, "Robust Scheduling against Selective Jamming in 6TiSCH Networks", draft-tiloca-6tisch-robust-scheduling-02 (work in progress), June 2019.

[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>.

[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/>.

[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, <https://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, <https://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, <https://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, <https://www.rfc-editor.org/info/rfc3444>.

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

[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, <https://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, <https://www.rfc-editor.org/info/rfc4291>.

[RFC4903] Thaler, D., "Multi-Link Subnet Issues", RFC 4903, DOI 10.17487/RFC4903, June 2007, <https://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, <https://www.rfc-editor.org/info/rfc4919>.

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

[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, <https://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, <https://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, <https://www.rfc-editor.org/info/rfc6347>.

[RFC6606] Kim, E., Kaspar, D., Gomez, C., and C. Bormann, "Problem Statement and Requirements for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing", RFC 6606, DOI 10.17487/RFC6606, May 2012, <https://www.rfc-editor.org/info/rfc6606>.

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

[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, <https://www.rfc-editor.org/info/rfc7426>.

[RFC8578] Grossman, E., Ed., "Deterministic Networking Use Cases", RFC 8578, DOI 10.17487/RFC8578, May 2019, <https://www.rfc-editor.org/info/rfc8578>.

[S-ALOHA] Roberts, L. G., "ALOHA Packet System With and Without Slots and Capture", doi 10.1145/1024916.1024920, April 1975, <https://dl.acm.org/citation.cfm?id=1024920>.

[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. Related Work In Progress

This document has been incremented as the work progressed following the evolution of the WG charter and the availability of dependent work. The intent was to publish when the WG concludes on the covered items. At the time of publishing the following specification are still in progress and may affect the evolution of the stack in a 6TiSCH-aware node.

A.1. Unchartered IETF work items

A.1.1. 6TiSCH Zerotouch security

The security model and in particular the zerotouch join process [I-D.ietf-6tisch-dtsecurity-zerotouch-join] depends on the ANIMA [ANIMA] Bootstrapping Remote Secure Key Infrastructures (BRSKI) [I-D.ietf-anima-bootstrapping-keyinfra] to enable zero-touch security provisionning; for highly constrained nodes, a minimal model based on pre-shared keys (PSK) is also available. As written to this day, it also depends on a number of documents in progress as CORE, and on "Ephemeral Diffie-Hellman Over COSE (EDHOC)" [I-D.selander-ace-cose-ecdhe], which is being considered for adoption at the LAKE WG.

A.1.2. 6TiSCH Track Setup

ROLL is now standardizing a reactive routing protocol based on RPL [I-D.ietf-roll-aodv-rpl] 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.

At the time of this writing, the formation of a new working group called RAW for Reliable and Available Wireless networking is being considered. The work on centralized Track computation is deferred to a subsequent work, not necessarily at 6TiSCH. A Predictable and Available Wireless (PAW) bar-BoF took place. RAW may form as a WG and develop a generic specification for Track operations that would cover 6TiSCH requirements as expressed in this architecture, more in [I-D.thubert-raw-technologies] and [I-D.pthubert-raw-problem-statement]. In a large LLN, it is not feasible to update the routes from a central controller that resides far over the constrained network at the speed at which the quality of the wireless links varies. RAW would focus on forwarding behaviors to react quickly and locally to the changes in the wireless links.

ROLL is also standardizing an extension to RPL to setup centrally-computed routes [I-D.ietf-roll-dao-projection]

The 6TiSCH Architecture should thus inherit from the DetNet [I-D.ietf-detnet-architecture] architecture and thus depends on it. The Path Computation Element (PCE) should be a core component of that architecture. An extension to RPL or to TEAS [TEAS] will be required to expose the 6TiSCH node capabilities and the network peers to the PCE, possibly in combination with [I-D.rahul-roll-mop-ext]. A protocol such as a lightweight PCEP or an adaptation of CCAMP [CCAMP] G-MPLS formats and procedures could be used in combination to [I-D.ietf-roll-dao-projection] to install the Tracks, as computed by the PCE, to the 6TiSCH nodes.

A.1.3. Using BIER in a 6TiSCH Network

ROLL is actively working on Bit Index Explicit Replication (BIER) as a method to compress both the dataplane packets and the routing tables in storing mode [I-D.thubert-roll-bier].

BIER could also be used in the context of the DetNet service layer. BIER-TE-based OAM, Replication and Elimination [I-D.thubert-bier-replication-elimination] leverages BIER Traffic Engineering (TE) to control in the data plane the DetNet Replication and Elimination activities, and to provide traceability on links where replication and loss happen, in a manner that is abstract to the forwarding information.

a 6loRH for BitStrings [I-D.thubert-6lo-bier-dispatch] proposes a 6LoWPAN compression for the BIER Bitstring based on 6LoWPAN Routing Header [RFC8138].

A.2. External (non-IETF) work items

The current charter positions 6TiSCH on IEEE Std. 802.15.4 only. Though most of the design should be portable on other link types, 6TiSCH has a strong dependency on IEEE Std. 802.15.4 and its evolution. The impact of changes to TSCH on this Architecture should be minimal to non-existent, but deeper work such as 6top and security may be impacted. A 6TiSCH Interest Group at the IEEE maintains the synchronization and helps 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.

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