

6TiSCH
Internet-Draft
Intended status: Informational
Expires: April 24, 2014

S. Chasko
L+G
S. Das
ACS
R. Marin-Lopez
University of Murcia
Y. Ohba, Ed.
Toshiba
P. Thubert
cisco
A. Yegin
Samsung
October 21, 2013

Security Framework and Key Management Protocol Requirements for 6TiSCH
draft-ohba-6tisch-security-00

Abstract

Since 6TiSCH forms layer 3 meshes over IPv6, use of key management protocols defined at layer 3 or above matches the target architecture so they can apply for the process by a new device of joining the mesh to extend it. This document details that particular operation within the whole 6TiSCH architecture.

Status of This Memo

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

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

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

This Internet-Draft will expire on April 24, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Acronyms	4
3. Security Framework	4
4. KMP requirements	7
4.1. Phase-1 KMP requirements	7
4.2. Phase-2 KMP requirements	8
5. Security Considerations	8
6. IANA Considerations	9
7. Acknowledgments	9
8. References	9
8.1. Normative References	9
8.2. Informative References	10
8.3. External Informative References	10
Appendix A. KMP candidates	11
A.1. Phase-1 KMP candidates	11
A.2. Phase-2 KMP candidates	11
Authors' Addresses	13

1. Introduction

The emergence of radio technology enabled a large variety of new types of devices to be interconnected, at a very low marginal cost compared to wire, at any range from Near Field to interplanetary distances, and in circumstances where wiring could be less than practical, for instance rotating devices.

At the same time, a new breed of Time Sensitive Networks is being developed to enable traffic that is highly sensitive to jitter and quite sensitive to latency. Such traffic is not limited to voice and video, but also includes command and control operations such as found in industrial automation or in-vehicular sensors and actuators.

6TiSCH aims at providing an open standard with new capabilities, both in terms of scalability (number of IPv6 devices in a single subnet) and in terms of guarantees (delivery and timeliness). Both the ISA100.11a and Wireless HART protocols are gaining acceptance in the automation industry and demonstrate that a level of determinism can be achieved on a wireless medium with adequate guarantees for low speed control loops, used in mission critical Process Control applications. For industrial applications, security is not an option and a power efficient authentication mechanism is strictly required.

For other usages such as rust control, intrusion detection or seismic activity monitoring, the capability to correlate inputs from multiple sources can be critical, and the value of the network directly augments with the number of connected devices. In order to scale to appropriate levels, the need for spatial reuse of the spectrum often implies routing capabilities over short range radios. Proprietary variations demonstrate that RPL can scale to multiple thousands of devices, but at the same time expose a new challenge for security that must enable deployments of any scale with security requirements that may vary widely. If the cost of the security in terms of network operations and system resources depends on that degree of security, then 6TiSCH should enable different profiles that can match different requirements and capabilities.

Since 6TiSCH forms layer 3 meshes over IPv6, key management protocols defined at layer 3 or above can apply for the process by a new device of joining the mesh to extend it. This document details that particular operation within the whole 6TiSCH architecture.

ZigBee IP [ZigBeeIP] ("ZigBee" is a registered trademark of the ZigBee Alliance) is a standard for IPv6-based wireless mesh networks using PANA for network access authentication and secure distribution of a link-layer group key called Network Key to authenticated mesh nodes formed over unslotted CSMA-CA MAC of 802.15.4. Each mesh node in the same ZigBee IP network derives the same link-layer key from the Network Key to protect IEEE 802.15.4 MAC frames exchanged between adjacent mesh nodes. While sharing the same link-layer key among all mesh nodes can make the required key state maintained by each mesh node compact, a compromise of a mesh node can lead to link-layer key leakage in the entire ZigBee IP network. Also, the cost of updating the link-layer key can be high as the key needs to be updated at all mesh nodes whenever the 4-octet frame counter at any single node wraps or the key is considered to be compromised or weak.

In the case of TSCH MAC which uses 5-octet global frame counter referred to as Absolute Slot Number (ASN), the frame counter is not likely to wrap in the expected lifetime of the device, but key update for a common link-layer key is still issue if the key needs to be changed for other reasons.

This document introduces a more secure and scalable key management framework for 6TiSCH networks and identifies requirements for key management protocols to be used in the framework.

2. Acronyms

In addition to the acronyms defined in [I-D.palattella-6tisch-terminology], the following acronyms are used in this document.

KMP: Key Management Protocol

PANA: Protocol for carrying Authentication for Network Access

SA: Security Association

MAC: Media Access Control

3. Security Framework

This section describes a security framework consisting of four phases as shown in Figure 1. The architecture is applicable to not only 6TiSCH networks but also non-time synchronized mesh networks. Each node in a mesh network runs through the following phases:

- o Phase-0 (Implanting Phase): In this phase, a node installs credentials used for subsequent phases in a physically secure and managed location before the node is placed to where it is expected to operate. Details on Phase-0 is outside the scope of this document.
- o Phase-1 (Bootstrapping Phase): In this phase, a node (re)installs credentials used for subsequent phases from an authentication server after it is placed to where it is expected to operate. The credentials installed during Phase-1 include Phase-2 credentials and Phase-3 credentials, and may also include long-term Phase-1 credentials if the initial Phase-1 credentials are intended for one-time use such as a temporary PIN. An authentication and key establishment protocol called a Phase-1 KMP is conducted between the node and the authentication server using Phase-1 credentials. The Phase-1 credentials have longer lifetime than Phase-2 and Phase-3 credentials so that Phase-2 and Phase-3 credentials can be

renewed using the Phase-1 credentials. Both symmetric and asymmetric key credentials can be used as Phase-1 credentials. In Phase-1 KMP, the Phase-2 and Phase-3 credentials are distributed from the authentication server to the node. When the authentication server is multiple hops away from the node, mutual authentication between the node and the authentication server is conducted via a neighboring node acting as an authentication relay. There may be no link-layer security available between the node and its neighboring node in this phase. An authentication server is typically (but is not necessarily) co-located with the coordinator of the mesh network. Phase-1 is optional if Phase-2 credentials are installed during Phase-0 and do not need to be updated.

- o Phase-2 (Link Establishment Phase): In this phase, the node performs mutual authentication with its neighboring node using the Phase-2 credentials to establish SAs between adjacent nodes for protecting 802.15.4 MAC frames. The authentication and key establishment protocol used in this phase is referred as a Phase-2 KMP or a link establishment KMP. For highly scalable mesh networks consisting of thousands of mesh nodes, certificates are used as the Phase-2 credentials. The SA of a link between node i and node j maintains link-layer keys, i.e., 128-bit keys used in AES-CCM* mode, a variant of the Counter with Cipher Block Chaining - Message Authentication Code (CBC-MAC) Mode, for encryption, authentication or authenticated encryption of 802.15.4 frames. K_i denotes a link-layer key for protecting broadcast MAC frames originated at node i . K_{ij} denotes a link-layer key for protecting unicast MAC frames originated at node i and destined for node j . There are several variations of forming link-layer keys.

1. $K_{ij}=K_i$ for all j
2. $K_{ij}=K_{ji}$, $K_i \neq K_j$ for all i, j ($i \neq j$)
3. $K_{ij} \neq K_{ji}$, $K_i \neq K_j$ for all i, j ($i \neq j$)

In model 1, unicast and broadcast keys for protecting MAC frames originated at a given node are the same. K_i and K_j may or may not be the same in model 1, and when $K_i=K_j=K$ for all i and j , then such K is considered as a common network key. In models 2 and 3, unicast and broadcast keys originated at a given node are distinct. The difference between models 2 and 3 is that unicast keys are bi-directional in model 2 while they are uni-directional in model 3. One model may be chosen among three depending on the required security level and the number of keys maintained by each node.

- o Phase-3 (Operational Phase): In this phase, the node is able to run various higher-layer protocols over IP over an established secure link. Additional authentication and key establishment may take place for the higher-layer protocols using Phase-3 credentials. A node in Phase-3 is able to process Phase-1 and Phase-2 KMPs. Example use cases are:
 - * A Phase-3 node can initiate a Phase-1 KMP to update its Phase-2 or Phase-3 credentials.
 - * A Phase-3 node can forward Phase-1 KMP messages originated from or destined for a Phase-1 node that is joining the mesh network through the Phase-3 node.
 - * A Phase-3 node can initiate a Phase 2 KMP to establish a new link with a newly discovered neighbor node.

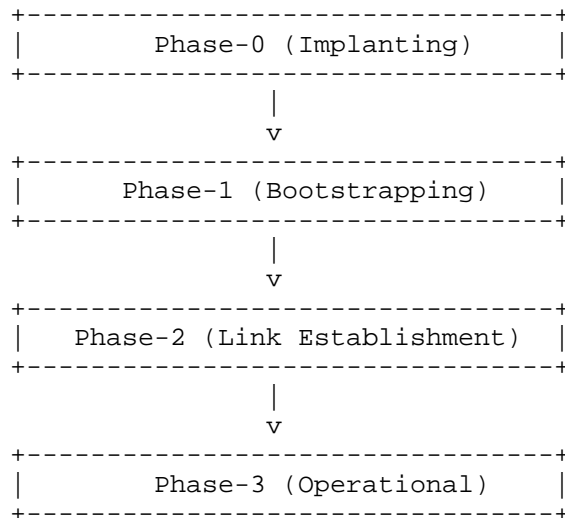
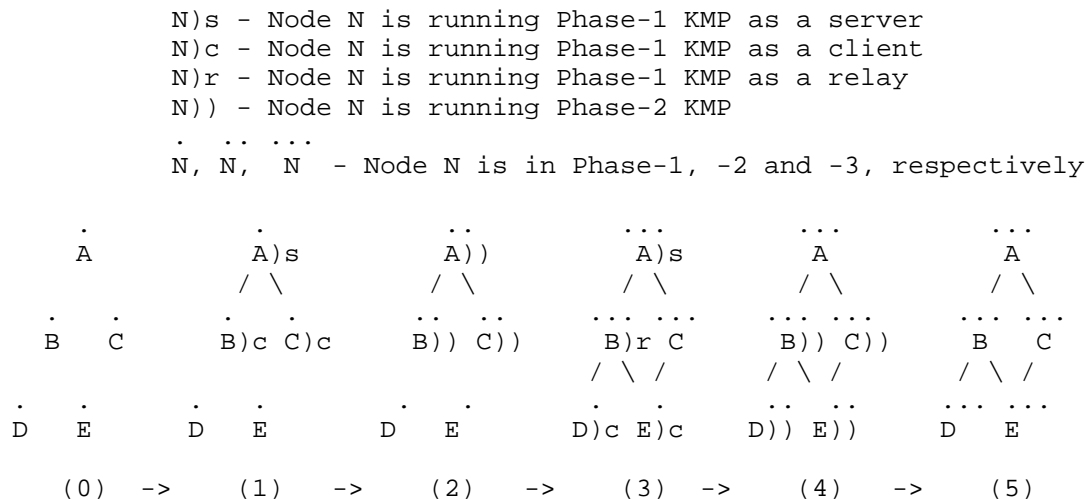


Figure 1: 4-Phase Key Management Model



(0) Initially all nodes are in Phase-1. (1) Nodes B and C run Phase-1 KMP with Node A (i.e., the authentication server) to obtain Phase-2 and Phase-3 credentials. (2) Nodes B and C run Phase-2 KMP with Node A. (3) Nodes D and E run Phase-1 KMP using Node B as an authentication relay. (Alternatively, Node E may use Node C as an authentication relay.) (4) Node D runs Phase-2 KMP with Node B. Node E runs Phase-2 KMP with Nodes B and C. (5) All nodes are operational.

Figure 2: Example Sequence

Since we already identified PANA as the Phase-1 KMP due to its authentication relay and secure credential distribution capabilities, and Phase-3 KMP requirements would depend on application protocols, we focus on Phase-2 KMP requirements in the next section.

4. KMP requirements

4.1. Phase-1 KMP requirements

Requirements on Phase-1 KMP are listed below.

R1-1: Phase-1 KMP MUST support mutual authentication.

R1-2: Phase-1 KMP MUST support stateless authentication relay operation.

R1-3:s Phase-1 KMP MUST support secure credential distribution.

4.2. Phase-2 KMP requirements

Requirements on Phase-2 KMP are listed below.

R2-1: Phase-2 KMP Nodes MUST mutually authenticate each other before establishing a link and forming a mesh network. No authentication server is involved in the Phase-2 authentication.

R2-2: Phase-2 KMP authentication credentials MAY be pre-provisioned or MAY be obtained via Phase-1 KMP.

R2-3: Phase-2 KMP authentication credentials MUST have a lifetime.

R2-4: Phase-2 KMP MUST support certificates for scalable operation.

R2-5: Phase-2 KMP message exchanges MUST be integrity and replay protected after successful authentication.

R2-6: Phase-2 KMP MUST have the capability to establish security association and unicast session keys after successful authentication to protect unicast MAC frames between nodes.

R2-7: Phase-2 KMP MUST have the capability to establish security association and broadcast session keys after successful authentication to protect broadcast MAC frames between nodes.

R2-8: Phase-2 KMP MUST support confidentiality to distribute the broadcast session keys securely.

5. Security Considerations

In this section, security issues that can potentially impact the operation of IEEE 802.15.4e TSCH MAC are described.

In TSCH MAC, time synchronization and channel hopping information are advertised in Enhanced Beacon (EB) frames [I-D.wattheyne-6tsch-tsch-lln-context]. The advertised information is used by mesh nodes to determine the timeslots available for transmission and reception of MAC frames. A rogue node can inject forged EB frames and can cause replay and DoS attacks to TSCH MAC operation. To mitigate such attacks, all EB frames MUST be integrity protected. While it is possible to use a pre-installed static key for protecting EB frames to every node, the static key becomes vulnerable when the associated MAC frame counter continues to be used after the frame counter wraps. Therefore, the 6TiSCH solution MUST provide a mechanism by which mesh nodes can use the available time slots to run Phase-1 and Phase-2 KMPs and provide integrity protection to EB frames.

6. IANA Considerations

There is no IANA action required for this document.

7. Acknowledgments

We would like to thank Thomas Watteyne, Jonathan Simon, Maria Rita Palattella and Rene Struik for their valuable comments.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5191] Forsberg, D., Ohba, Y., Patil, B., Tschofenig, H., and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA)", RFC 5191, May 2008.
- [RFC6345] Duffy, P., Chakrabarti, S., Cragie, R., Ohba, Y., and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA) Relay Element", RFC 6345, August 2011.
- [RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012.
- [RFC6786] Yegin, A. and R. Cragie, "Encrypting the Protocol for Carrying Authentication for Network Access (PANA) Attribute-Value Pairs", RFC 6786, November 2012.
- [I-D.palattella-6tisch-terminology]
Palattella, M., Thubert, P., Watteyne, T., and Q. Wang, "Terminology in IPv6 over the TSCH mode of IEEE 802.15.4e", draft-palattella-6tisch-terminology-00 (work in progress), October 2013.
- [I-D.watteyne-6tsch-tsch-lln-context]
Watteyne, T., Palattella, M., and L. Grieco, "Using IEEE802.15.4e TSCH in an LLN context: Overview, Problem Statement and Goals", draft-watteyne-6tsch-tsch-lln-context-02 (work in progress), May 2013.
- [I-D.moskowitz-hip-rg-dex]
Moskowitz, R., "HIP Diet EXchange (DEX)", draft-moskowitz-hip-rg-dex-06 (work in progress), May 2012.

8.2. Informative References

- [RFC4137] Vollbrecht, J., Eronen, P., Petroni, N., and Y. Ohba, "State Machines for Extensible Authentication Protocol (EAP) Peer and Authenticator", RFC 4137, August 2005.
- [RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, September 2007.
- [RFC5705] Rescorla, E., "Keying Material Exporters for Transport Layer Security (TLS)", RFC 5705, March 2010.
- [RFC6550] Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, March 2012.
- [I-D.keoh-tls-multicast-security]
Keoh, S., Kumar, S., and E. Dijk, "DTLS-based Multicast Security for Low-Power and Lossy Networks (LLNs)", draft-keoh-tls-multicast-security-00 (work in progress), October 2012.
- [I-D.ietf-hip-rfc5201-bis]
Moskowitz, R., Heer, T., Jokela, P., and T. Henderson, "Host Identity Protocol Version 2 (HIPv2)", draft-ietf-hip-rfc5201-bis-14 (work in progress), October 2013.
- [I-D.draft-palattella-6tisch-terminology]
Palattella, MR., Ed., Thubert, P., Watteyne, T., and Q. Wang, "Terminology in IPv6 over Time Slotted Channel Hopping. draft-palattella-6tisch-terminology-00 (work in progress) ", March 2013.
- [I-D.draft-thubert-6tisch-architecture]
Thubert, P., Ed., Assimiti, R., and T. Watteyne, "An Architecture for IPv6 over Time Synchronized Channel Hopping. draft-thubert-6tisch-architecture-00 (work in progress) ", March 2013.

8.3. External Informative References

- [IEEE802154e]
IEEE standard for Information Technology, "IEEE std. 802.15.4e, Part. 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANS) Amendment 1: MAC sublayer", April 2012.

[IEEE802154]

IEEE standard for Information Technology, "IEEE std. 802.15.4, Part. 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks", June 2011.

[ZigBeeIP]

ZigBee Public Document 15-002r00, "ZigBee IP Specification", 2013.

Appendix A. KMP candidates

A.1. Phase-1 KMP candidates

PANA [RFC5191] is the Phase-1 KMP candidate since it supports mutual authentication, stateless authentication relay function [RFC6345] and encrypted distribution of attributes [RFC6786]. The PANA Authentication Agent (PAA) is located in the coordinator of the mesh network.

A.2. Phase-2 KMP candidates

Once Phase-1 is complete by using PANA, it is assumed that node will have a certified public key (and associated private key). A candidate Phase 2 KMP must use this certified public key to perform an authentication process. As a consequence of a successful authentication some cryptographic material for unicast and multicast link protection between nodes must be generated.

A list of candidate protocols may provide the requirements defined in Section 4.2 (this is a preliminary list that may be extended in the future):

- o HIP DEX [I-D.moskowitz-hip-rg-dex]. The Host Identity Protocol Diet EXchange (HIP DEX) is a lighter version of the HIP Base Exchange (HIP BEX) [I-D.ietf-hip-rfc5201-bis] specifically designed to be used in constrained devices (e.g., sensor networks). In particular, HIP DEX may be used to authenticate two IEEE 802.15.4 nodes and provide key material for a MAC layer security protocol as supported in IEEE 802.15.4. However, by just using the value of the public key and the private key is not enough to carry out the authentication between nodes. In particular, a node A and node B should not be able to successfully finish HIP DEX execution if they both have not been authenticated in Phase-1. Thus, HIP DEX will require the inclusion of the certificate of each node to achieve full mutual authentication. The information in the certificate must ensure that the node was authenticated in Phase-1. In consequence, HIP DEX must include a

CERT parameter for carrying this certificate. Once the HIP DEX protocol has successfully finished a Pair-Wise Key SA is derived. This SA is used to secure and authenticate user data, thus it can be used to provide the required keys for protecting IEEE 802.15.4 unicast MAC frames. The same message is used to refresh the Pair-Wise Key SA. So far HIP DEX only specifies how this key material is used for protecting data traffic with ESP. To distribute multicast keys HIP DEX may also use UPDATE message. For less resource-constrained devices, HIP-BEX (Basic Exchange) is more suitable than HIP-DEX since HIP-BEX has better security properties (such as use of ephemeral Diffie-Hellman) than HIP-DEX at the cost of increased complexity.

- o PANA [RFC5191] and some certificate-based EAP method. Another candidate is to use PANA between node A and node B. In this case, one of the nodes (e.g. node A) acts as PaC while the other (e.g. node B) is acting as PAA. Moreover the PAA will operate in standalone mode [RFC4137]. That is, the EAP server is placed on the PAA and not in a backend authentication server. Finally, the selected EAP method must work with public key/private key cryptography. Once the PAA authentication is complete, the PaC and PAA can derive cryptographic material (for instance, from the MSK) which can be used to protect unicast MAC frames. Furthermore, by using the extension defined in [RFC6345] is possible to distribute a multicast key encrypted with the PANA SA. It is worth noting that, though this candidate solution leverages the PaC implementation from Phase-1, each node needs to deploy a PAA implementation, an EAP server and a specific EAP method, which may be different from the one used for Phase-1.

- o DTLS [RFC6347]. Datagram Transport Layer Security (DTLS) is being considered in constrained devices for protecting application data traffic (e.g. CoAP). It is not only being considered for unicast application data traffic but also for multicast data traffic [I-D.keoh-tls-multicast-security]. In particular, a multicast key is distributed over an unicast DTLS channel established between two nodes (node A and node B). This multicast key is used to protect multicast traffic by using TLS records. The Phase2-KMP should be able to export this key material to the IEEE 802.15.4 MAC layer so that the protection is carried out at link layer. In [RFC5705], a mechanism for exporting key material after a TLS/DTLS execution is defined. Nevertheless, the exported key material is intended to be used in unicast communications for upper layers or protocols at upper layers. However, a mechanism for exporting multicast key is not specified. In principle, this exported key material may be used for protecting unicast IEEE 802.15.4 MAC frames. However, this usage and multicast key management using DTLS for multicast IEEE 802.15.4 protection need further investigation.

Authors' Addresses

Stephen Chasko
Landis+Gyr
3000 Mill Creek Ave.
Alpharetta, GA 30022
USA

Email: Stephen.Chasko@landisgyr.com

Subir Das
Applied Communication Sciences
1 Telcordia Drive
Piscataway, NJ 08854
USA

Email: sdas@appcomsci.com

Rafa Marin-Lopez
University of Murcia
Campus de Espinardo S/N, Faculty of Computer Science
Murcia 30100
Spain

Phone: +34 868 88 85 01
Email: rafa@um.es

Yoshihiro Ohba (editor)
Toshiba Corporate Research and Development Center
1 Komukai-Toshiba-cho
Saiwai-ku, Kawasaki, Kanagawa 212-8582
Japan

Phone: +81 44 549 2127
Email: yoshihiro.ohba@toshiba.co.jp

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

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

Alper Yegin
Samsung
Istanbul
Turkey

Email: alper.yegin@yegin.org

6TiSCH
Internet-Draft
Intended status: Informational
Expires: April 14, 2014

MR. Palattella, Ed.
SnT/Univ. of Luxembourg
P. Thubert
cisco
T. Watteyne
Linear Technology / Dust Networks
Q. Wang
Univ. of Sci. and Tech. Beijing
October 11, 2013

Terminology in IPv6 over the TSCH mode of IEEE 802.15.4e
draft-palattella-6tisch-terminology-00

Abstract

6TiSCH proposes an architecture for an IPv6 multilink subnet that is composed of a high speed powered backbone and a number of IEEE802.15.4e TSCH wireless networks attached and synchronized by backbone routers. This document extends existing terminology documents available for Low-power and Lossy Networks to provide additional terminology elements.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on April 14, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Terminology	3
3. IANA Considerations	8
4. Security Considerations	8
5. Acknowledgements	8
6. References	8
6.1. Normative References	8
6.2. Informative References	9
6.3. External Informative References	10
Authors' Addresses	10

1. Introduction

A new breed of Time Sensitive Networks is being developed to enable traffic that is highly sensitive to jitter and quite sensitive to latency. Such traffic is not limited to voice and video, but also includes command and control operations such as in industrial automation or in-vehicle sensors and actuators.

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

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

2. Terminology

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

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

This document adds the following terms:

- 6TiSCH:** IPv6 over the Timeslotted Channel Hopping (TSCH) mode of IEEE 802.15.4e. It defines a set of IETF sublayers and protocols (in particular, for setting up a schedule with a centralized or distributed approach, managing the resource allocation), as well as the architecture to bind them together, for use in IPv6 TSCH based networks.
- 6F:** IPv6 Forwarding. One of the three forwarding model supported by 6TiSCH. Packets are routed at layer 3, where QoS and RED operations are expected to prioritize flows with differentiated services.
- 6top:** 6top is the adaptation layer between TSCH and upper layers like 6LoWPAN and RPL. It is defined in [I-D.draft-wang-6tsch-6top].
- 6top Data Convey Model:** Model describing how the 6top adaptation layer feeds the data flow coming from upper layers into TSCH. It is composed by an I-MUX module, a MUX module, a set of priority queues, and a PDU (Payload Data Unit).
- ASN:** Absolute Slot Number, the timeslot counter, incremented by one at each timeslot. It is wide enough to not roll over in practice. See [I-D.watteyne-6tsch-tsch-lln-context].
- Blacklist:** Set of frequencies which should not be used for communication.
- BBR:** Backbone Router. In the 6TiSCH architecture, it is an LBR and also a NEAR. It performs ND proxy operations between registered devices and classical ND devices that are located over the backbone.
- Bundle:** A group of equivalent scheduled cells, i.e. cells identified by different [slotOffset, channelOffset],

which are scheduled for a same purpose, with the same neighbor, with the same flags, and the same slotframe. The size of the bundle refers to the number of cells it contains. Given the length of the slotframe, the size of the bundle translates directly into bandwidth, either logical, or physical.

Cell: A single element in the TSCH slotframe, identified by a slotOffset value, a channelOffset value, a slotframe_ID and Hopping_Sequence_ID. A cell can be scheduled or unscheduled. During an unscheduled cell, the node does not communicate. When a cell is scheduled, it is assigned a MAC-layer slotframe identifier, a neighbor MAC address (which can be the broadcast address), and one or more of the following flags: TX, RX, shared, timeskeeping, hard. A broadcast cell is an alias for "a scheduled cell with neighbor address the broadcast address".

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

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

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

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

- Distributed track reservation: A reservation of a track done by one or more in-network entities (typically a connection endpoint).
- EB: Enhanced Beacon frame used by an advertising node to announce the presence of the network. It contains information about the timeslot length, the current ASN value, the slotframes and timeslots the beaconing mote is listening on, and a 1-byte join priority (i.e., number of hops separating the node sending the EB, and the PAN coordinator).
- FF: 6LoWPAN Fragment Forwarding. It is one of the three forwarding model supported by 6TiSCH. The 6LoWPAN Fragment is used as a label for switching at the 6LoWPAN sublayer, as defined in [I-D.thubert-roll-forwarding-frags].
- GMPLS: Generalized Multi-Protocol Label Switching, a 2.5 layer service that is used to forward packets based on the concept of generalized labels.
- Hard Cell: A scheduled cell that is locked, i.e., it cannot be moved by 6top in the schedule. See [I-D.draft-wang-6tsch-6top].
- Hopping Sequence: Sequence of frequencies, identified by a Hopping_Sequence_ID, used for channel hopping, when translating the channel offset value into a frequency (i.e., PHY channel). See [I-D.watteyne-6tsch-tsch-lln-context].
- IE: Information Elements, a list of Type-Length-Value containers placed at the end of the MAC header, used to pass data between layers or devices. A small number of types are defined by TSCH, but a range of types is available for extensions, and thus, is exploitable by 6TiSCH. See [I-D.watteyne-6tsch-tsch-lln-context].
- I-MUX module: Inverse-Multiplexer, a classifier that receives 6LoWPAN frames and places them into priority queues.
- Interaction Model: It is a particular way of implementing a communication paradigm. Defined at a lower level of abstraction, it includes protocol-specific details such as a particular method (e.g., a REST GET) and a Data Model for the state to be exchanged.

- KMP: Key Managment Protocol.
- LBR: LLN Border Router. It is an LLN device, usually powered, that acts as a Border Router to the outside within the 6TiSCH architecture.
- Link: A communication facility or medium over which nodes can communicate at the link layer, i.e., the layer immediately below IP. Thus, the IETF parlance for the term "Link" is adopted, as opposed to the incompatible IEEE802.15.4e terminology. In the context of the 6TiSCH architecture, which applies to Low Power Lossy Networks (LLNs), an IPv6 subnet is usually not congruent to a single link and techniques such as IPv6 Neighbor Discovery Proxying and Routing Over LLNs are required to achieve reachability within the multilink subnet. A link is distinct from a track. In fact, link local addresses are not expected to be used over a track for end to end communication. Finally, from the Layer 3 perspective (where the inner complexities of TSCH operations are hidden to enable classical IP routing and Forwarding), a single radio interface may be seen as a number of Links with different capabilities for unicast or multicast services.
- Logical Cell: A cell that corresponds to granted bandwidth but is only lazily associated to a physical cell, based on usage.
- MAC: Medium Access Control.
- MUX module: Multiplexer, the entity that dequeues frames from priority queues and associates them to a cell for transmission.
- NEAR: Energy Aware Default Router, as defined in [I-D.chakrabarti-nordmark-6man-efficient-nd].
- NME: Network Management Entity, the entity in the network managing cells and other device resources. It may cooperate with the PCE. It interacts with LLN nodes through the backbone router.
- PANA: Protocol for carrying Authentication for Network Access, as defined in [RFC5191] . It is the protocol used in the 6TiSCH architecture for handling authentication during the join process.

- PCE: Path Computation Element, the entity in the network which is responsible for building and maintaining the TSCH schedule, when centralized scheduling is used.
- PCE cell reservation: The reservation of a cell done by the PCE.
- PCE track reservation: The reservation of a track done by the PCE.
- QoS: Quality of Service.
- SA: Security Association.
- Shared Cell: A cell that is used by more than one transmitter nodes at the same time and on the same channelOffset. Only cells with TX flag can be marked as "shared". A backoff algorithm is used to resolve contention.
- SlotOffset: Identifies a column in the TSCH schedule, i.e., the number of timeslots since the beginning of the current iteration of the slotframe.
- Slotframe: A MAC-level abstraction that is internal to the node and contains a series of timeslots of equal length and priority. It is characterized by a slotframe_ID, and a slotframe_size. Multiple slotframes can coexist in a node's schedule, i.e., a node can have multiple activities scheduled in different slotframes, based on the priority of its packets/traffic flows. The timeslots in the Slotframe are indexed by the SlotOffset; the first timeslot is at SlotOffset 0.
- Soft Cell: A scheduled cell that is not locked, i.e., it may be moved in the schedule within a same slotframe by 6top, as described in [I-D.draft-wang-6tsch-6top].
- TF: Track Forwarding. It is the simplest and fastest forwarding model supported by 6TiSCH. It is a G-MPLS-like forwarding model. The input cell characterises the flow and indicates the output cell.
- Timeslot: A basic communication unit in TSCH which allows a transmitter node to send a frame to a receiver neighbor, and that receiver neighbor to optionally send back an acknowledgment. The length of the timeslot determines the maximum size of the frame that can be exchanged.

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

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

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

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

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

3. IANA Considerations

This specification does not require IANA action.

4. Security Considerations

This specification is not found to introduce new security threat.

5. Acknowledgements

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

6. References

6.1. Normative References

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

- [RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", RFC 3444, January 2003.
- [RFC5191] Forsberg, D., Ohba, Y., Patil, B., Tschofenig, H., and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA)", RFC 5191, May 2008.
- [RFC6550] Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, March 2012.
- [RFC6552] Thubert, P., "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6552, March 2012.

6.2. Informative References

- [I-D.chakrabarti-nordmark-6man-efficient-nd]
Chakrabarti, S., Nordmark, E., Thubert, P., and M. Wasserman, "Efficiency aware IPv6 Neighbor Discovery Optimizations", draft-chakrabarti-nordmark-6man-efficient-nd-02 (work in progress), July 2013.
- [I-D.draft-sudhaakar-6tisch-coap]
Sudhaakar, R., Ed. and P. Zand, "6TiSCH Data Model for CoAP-00 (work in progress) ", October 2013.
- [I-D.draft-wang-6tsch-6top]
Wang, Q., Ed., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top). draft-wang-6tisch-6top-00 (work in progress) ", October 2013.
- [I-D.ietf-roll-terminology]
Vasseur, J., "Terms used in Routing for Low power And Lossy Networks", draft-ietf-roll-terminology-13 (work in progress), October 2013.
- [I-D.ohba-6tsch-security]
Chasko, S., Das, S., Lopez, R., Ohba, Y., Thubert, P., and A. Yegin, "Security Framework and Key Management Protocol Requirements for 6TSCH", draft-ohba-6tsch-security-01 (work in progress), July 2013.
- [I-D.thubert-6tisch-architecture]
Thubert, P., Assimiti, R., and T. Watteyne, "An Architecture for IPv6 over the TSCH mode of IEEE

IEEE802.15.4e", draft-thubert-6tisch-architecture-00 (work in progress), October 2013.

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

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

[I-D.vilajosana-6tisch-minimal]

Vilajosana, X. and K. Pister, "Minimal 6TiSCH Configuration", draft-vilajosana-6tisch-minimal-00 (work in progress), October 2013.

[I-D.watteyne-6tsch-tsch-lln-context]

Watteyne, T., Palattella, M., and L. Grieco, "Using IEEE802.15.4e TSCH in an LLN context: Overview, Problem Statement and Goals", draft-watteyne-6tsch-tsch-lln-context-02 (work in progress), May 2013.

6.3. External Informative References

[IEEE802154e]

IEEE standard for Information Technology, "IEEE std. 802.15.4e, Part. 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer", April 2012.

Authors' Addresses

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

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

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

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

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

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

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

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

6TiSCH
INTERNET-DRAFT
Intended Status: Informational
Expires: April 21, 2014

G. Piro
(Politecnico di Bari)
G. Boggia
(Politecnico di Bari)
L. A. Grieco
(Politecnico di Bari)
October 18, 2013

A standard compliant security framework for Low-power and Lossy Networks
draft-piro-6tisch-security-issues-00

Abstract

The aim of this Internet Draft is to define a standard compliant security framework for Low-power and Lossy Networks, in order to enable the possibility to encrypt and authenticate messages exchanged by nodes at the MAC layer. The framework is fully compatible with both IEEE 802.15.4 and IEEE 802.15.4e standards and offers a wide range of security features to network architectures developed within the 6TiSCH Working Group. In particular, it offers: (i) different kinds of security architectures; (ii) an efficient mechanism for initializing a secure IEEE 802.15.4 domain; and (iii) a lightweight mechanism to negotiate link keys among devices.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

Copyright and License Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1	Acronyms	3
2	Introduction	3
3	Security in IEEE 802.15.4	5
4	Definition of the secured domain	8
5	Security Configurations	8
5.1	Minimum security requirements	9
6	Initialization of a secured IEEE 802.15.4 domain	11
6.1	Setting-up phase	11
6.2	Bootstrap phase for a FFD device	13
6.3	Bootstrap phase for a RFD device in a Beacon-enabled network	14
6.4	Bootstrap phase for a RFD device in a not-Beacon-enabled network	15
6.5	Key negotiation phase	16
6.5.1	Key exchange mechanism based on DH	18
6.5.2	Generation of the LinkKeys	21
6.5.3	Update of MAC security attribute for the FFD node after the generation of the LinkKey	21
6.5.4	Update of MAC security attribute for the RFD node after the generation of the LinkKey	23
7	Additional features	23
8	Security Considerations	25
9	IANA Considerations	25
10	References	25
10.1	Normative References	25
10.2	Informative References	25
	Authors' Addresses	25

1 Acronyms

The following acronyms are used in this document:

DH Diffie Hellman

DEX HIP Diet EXchange

DTLS Datagram Transport Layer

LLN Low-power and Lossy Network

LBR LLN Border Routers

FFD Full Function Device

HIP Host Identity Protocol

MAC Medium Access Control

PAN Personal Area Network

PANA Protocol for Carrying Authentication for Network Access

PHY Physical

RFD Reduced Function Device

RSA Rivest-Shamir-Adleman

TSCH Time-slotted Channel Hopping

2 Introduction

The IEEE 802.15.4 standard [IEEE802154] is widely recognized as one of the most successful enabling technologies for short-range low-rate wireless communications. It covers all the details related to the Medium Access Control (MAC) and Physical (PHY) layers of the protocol stack and supports the possibility to protect MAC packets by means of symmetric-key cryptography techniques with several security options.

This standard relies on upper layers to orchestrate, enable, configure, and negotiate security services, as well as to handle the creation and exchange of encryption keys. But it leaves open some aspects, such as the initialization of a secure IEEE 802.15.4 domain, the generation and the exchange of keys, the management of joining operations in a secure 802.15.4 network already configured in the

past.

The IEEE 802.15.4e standard introduces some amendments to the IEEE 802.15.4 standard. The most important one is the Time-slotted Channel Hopping (TSCH), i.e., a novel MAC protocol, which better supports multi-hop communications in emerging industrial applications.

Since the IEEE 802.15.4e amendment focuses only on link-layer aspects, the 6TiSCH Working Group was born to suggest the adoption of IPv6 over the TSCH, thus covering all facets related to the schedule of network communications in complex (and eventually distributed) Low-Power and Lossy Networks (LLNs).

In industrial environments, security issues are very important. Hence, a complete framework, which supports a wide range of security features, is highly required.

In this context, the security in LLNs has been firstly addressed within ZigBee IP specifications, i.e., a suite of high level communication protocols sitting on top of the IEEE 802.15.4 MAC [ZIGBEEIP]. ZigBee IP supports end-to-end and link-layer security and a public key infrastructure based on X.509 certificates. It imposes the adoption of the same link-key (shared among all nodes) to protect packets belonging to any kind of services (i.e., only a single security level is allowed). This makes the network highly sensible to the presence of compromised devices. Moreover, the cost needed to update the key within the network could be high, especially for heavy loaded systems.

Security aspects have been also faced in [6TSCH-SEC] and [GARCIA-SEC-IOT]. The work [6TSCH-SEC] introduces a simple security framework based on four consecutive phases (i.e., implanting, bootstrapping, link establishment, and operational phases) which allow the initialization of a secured IEEE 802.15.4 network. It exploits well-known approaches, like PANA [RFC5191], HIP DEX [HIPDEX], and DTLS [RFC6347], to authenticate nodes and to negotiate link keys. Instead, [GARCIA-SEC-IOT] defines a set of security profiles to guarantee in different scenarios (e.g., network without security requirements, home, managed home, industrial, and advanced industrial). For each scenario, it identifies a number of security threats and suggests how fixing them through well-known protocols and algorithms.

Proposals presented in [6TSCH-SEC] and [GARCIA-SEC-IOT] do not provide a complete definition of a security framework for LLNs, which offers, at the same time, an accurate procedure for the network initialization, the support for different security configurations, and a fully compatibility with the standard. In addition, they do not guarantee the real possibility to implement conceived proposals in

devices, i.e., sensors with very limited computational, energy, and storage capabilities.

This Internet Draft proposes a complete and standard compliant framework offering a number of security features at the MAC layer of LLNs. All aspects have been designed in order to ensure an easy and effectively implementation on real devices.

In particular, the security framework described in this document introduces all the details required to enable the security for both IEEE 802.15.4 and IEEE 802.15.4e networks, as well as the possibility to encrypt and authenticate any kind of communications devised by the 6TiSCH Working Group.

In summary, the security framework covers: (1) five different kinds of security configurations; (2) an efficient mechanism for initializing a secure IEEE 802.15.4 (or IEEE 802.15.4e) domain; (3) a lightweight mechanism to negotiate link keys among devices; (4) a very simple technique adopted to update link keys during the time.

3 Security in IEEE 802.15.4

This section summarizes security features defined within the standard [IEEE802154], thus simplifying the comprehension of the remaining part of this Internet Draft.

The IEEE 802.15.4 standard defines eight security levels to protect MAC frames, as summarized in Fig. 1.

Security level	Security attribute	Data Integrity	Data Confidentiality
0	None	No	No
1	MIC-32	Yes	No
2	MIC-64	Yes	No
3	MIC-128	Yes	No
4	ENC	No	Yes

5	ENC-MIC-32	Yes	Yes	
+-----+	+-----+	+-----+	+-----+	+-----+
6	ENC-MIC-64	Yes	Yes	
+-----+	+-----+	+-----+	+-----+	+-----+
7	ENC-MIC-128	Yes	Yes	
+-----+	+-----+	+-----+	+-----+	+-----+

Figure 1. Security levels available for a IEEE 802.15.4 network.

At the MAC layer, encryption and decryption functionalities are implemented within the "outgoing frame security" and the "incoming frame security" procedures, respectively. They exploits a number of security attributes, summarized in what follows:

- macKeyTable: it is composed by a set of KeyDescriptor elements. A specific KeyDescriptor element is created for each key, composed by (see Tab. 61 of the IEEE 802.15.4 standard for more details [IEEE802154]):
 - The KeyIdLookupList, which is a list of KeyIdLookupDescriptor entries. A KeyIdLookupDescriptor is composed by a set of parameters (see Tab. 65 of the IEEE 802.15.4 standard for more details [IEEE802154]), i.e., KeyIdMode, KeySource, KeyIndex, DeviceAddMode, DevicePANId, and DeviceAddress, that are used to identify the key within the macKeyTable.
 - The DeviceDescriptorHandleList, which contains pointers to DeviceDescriptor elements stored within the macDeviceTable. It is used to identify which devices may use the key.
 - The KeyUsageList, which is a list of KeyUsageDescriptor elements. A KeyUsageDescriptor is composed by the FrameType and the CommandFrameIdentifies fields that indicate the frame type with which the considered key may be used (see Tab. 62 of the IEEE 802.15.4 standard for more details [IEEE802154]).
 - The Key.
- macDeviceTable: it is composed by a set of DeviceDescriptor elements, providing some information about remote devices which the node can establish a secure communication with. A dedicated DeviceDescriptor element is associated to each remote device. It is composed by a number of fields, i.e., PANId, ShortAddress, ExtAddress, FrameCounter, and Extemp, which collect information related to a specific remote device (see Tab. 64 of the IEEE 802.15.4 standard for more details [IEEE802154]).

- macSecurityLevelTable: it is made by a set of SecurityLevelDescriptor elements, which store details about the security level required for each MAC frame type and subtype. Fields belonging to the SecurityLevelDescriptor data structure are: FrameType, ComamndFrameIdentifier, SecurityMinimum, DeviceOverrideSecurityMinimum, and AllowedSecurityLevels (see Tab. 63 of the IEEE 802.15.4 standard for more details [IEEE802154]).
- macFrameCounter: it is an integer value storing the outgoing frame counter for the considered device.
- macAutoRequestSecurityLevel: it is an integer value providing the security level used for automatic data requests.
- macAutoRequestKeyIdMode: it is an integer value indicating the key identifier mode used for automatic data requests. It is not valid if the macAutoRequestSecurityLevel attribute is set to 0x00.
- macAutoRequestKeySource: it represents a short or extended IEEE 802.15.4 MAC address, indicating the originator of the key used for automatic data requests. This attribute is not valid if the macAutoRequestKeyIdMode element is not valid or set to 0x00.
- macAutoRequestKeyIndex: it is an integer value storing the index of the key used for automatic data requests. It is not valid if the macAutoRequestKeyIdMode attribute is not valid or set to 0x00.
- macDefaultKeySource: it is the extended IEEE 802.15.4 MAC address of the originator of the default key used for key identifier mode 0x01.

During the outgoing security procedure, the high layer uses the KeyIdMode parameter to select a specific key in the macKeyTable to be used for protecting the MAC frame.

The KeyIdMode is set to 00, 01, 10, and 11 in the case the key can be derived implicitly by both sender and the receiver and its is not specified in the message, the key is determined explicitly by the KeyIndex parameter stored into the MAC header and the macDefaultKeySource, the key can be derived by considering KeyIndex and KeySource fields stored into the MAC header (with KeySource representing the short address of the device that has generated the key), and the key can be derived by considering KeyIndex and KeySource fields stored into the MAC header (with KeySource representing the IEEE extended address of the device that has generated the key), respectively.

The IEEE 802.15.4 standard does not provide any guideline to create (and or negotiate) keys, as well as to configure the aforementioned security MAC attributes.

4 Definition of the secured domain

In this document, the "secured domain" concept refers to the portion of a LLN network where procedures and techniques described in this proposal must be performed in order to configure and maintain secured communications.

Two types of network nodes, i.e., the Full Function Device (FFD) and the Reduced Function Device (RFD), can be found in an IEEE 802.15.4 network. They can be arranged in both peer-to-peer and star topologies. A FFD device has the highest computational capabilities and it works as the coordinator of the network (also called PAN coordinator), thus being a reference node for a group of other RFD devices. Instead, a RFD device has lower resource and communication capabilities and it is able to communicate only with its reference FFD. Whereas RFD nodes must be connected only to coordinator, FFD devices can connect among them forming more complex meshed network architectures.

The concept of secured domain adopted in this Internet Draft coincides with the broadcast domain of an IEEE 802.15.4 network, which is composed by a number of RFD connected to a coordinator and the coordinator itself.

The IEEE 802.15.4e amendment does not introduces any variations to the network architecture, thus the previous definition is still valid in this context.

The same consideration can be done for network architectures and models designed within the 6TiSCH Working Group that, as explained before, are based on top of the IEEE 802.15.4e amendment.

Definitively, to the sake of clarity, from this moment on, we will focus on the terminology related to the IEEE 802.14.5 standard, implying that the entire framework can be adopted also for IEEE 802.15.4e networks and 6TiSCH's proposals.

5 Security Configurations

The security framework described in this document supports five network configurations, which differ among them according to the offered security features. They are:

- Fully Secured network: all the IEEE 802.15.4 devices forming the network are configured to fully support security services. It represents the most secured configuration: all packets, independently from the message they carry, are encrypted and authenticated. Nodes that do not support security capabilities (or that are not in possession of all the information to joining the network, such as key materials and encryption and decryption algorithms) are not allowed to join the network.
- Unsecured network: security services are not supported. Even if in possession of security capabilities, any pair of nodes is not allowed to establish a secured communication. Differently for the Fully Secured scheme, this is offers the lowest security level. Since the data encryption, the message integrity, and the peer authentication are not implemented, all the MAC frames are exchanged in clear. Hence, the setup and the maintaining of the network are described by the standard and no further upgrades are required.
- Partial Secured network: only the integrity of message is supported.
- Hybrid Secured network: the network can be composed by heterogeneous nodes that could or could not support security features. As default, the network is created in an unsecured manner. All the non-unicast control messages sent by the coordinator should be transmitted in clear. In this way, in fact, it is ensured that all the devices are able to read the content of packets. A RFD node with security capabilities, that intends to exchange encrypted and/or authenticated packets with the coordinator, could negotiate a set of link key with its reference FFD.
- Flexible Secured network: as default, the network is setup with the Fully Secured configuration and all packets are encrypted and authenticated. If there is at least one node that have not security capabilities, the coordinator could decides to switch to the Hybrid Secured configuration.

5.1 Minimum security requirements

The IEEE 802.15.4 standard imposes to specify, for each kind of MAC packet, minimum security levels that should be guaranteed. These restrictions must be detailed for each remote device.

To this end, SecurityMinimum, DeviceOverrideSecurityMinimum, and AllowedSecurityLevels parameters are stored into the DeviceDescriptor element (see Sec. 3) to define the minimum security level (i.e., one of those reported in Fig.1), the possibility to override the minimum security level (i.e., DeviceOverrideSecurityMinimum is just a boolean flag), and the list of allowed security levels in the case the minimum one could be overridden, respectively.

With reference to secured network configurations presented in Sec. 3.1, these parameters must be set as reported in Fig. 2.

Attribute	Secured Network Configurations				
	Unsecured	Fully	Partial	Hybrid	Flexible
SecurityMinimum	0	from 5 to 7	from 1 to 4	0	from 1 to 7
DeviceOverride- SecurityMinimum	FALSE	FALSE	FALSE	FALSE	TRUE
AllowedSecuri- tyLevelsvels	0	from 5 to 7	from 1 to 4	from 0 to 7	from 0 to 7

Figure 2. Setting of security attributes of the DeviceDescriptor element in each proposed secure network configuration.

The Unsecured network configuration does not support any security features. Hence, both minimum and allowable security levels are set to 0 for all the MAC frames and the possibility to override such constraints is disabled for all devices.

If the Fully Secured configuration is enabled, the minimum security level must be chosen in the range [5,7], thus allowing the possibility to support the encryption and the authentication of messages. The manufacturer must set the default value to 7; it can be updated by the network administrator. The minimum security level must not be overridden by any devices and, as a consequence, the field AllowedSecurityLevels should contain only one value, equal to the minimum security level.

If the Partial Secured configuration is enabled, the minimum security level must be chosen in the range [1,4], thus allowing the possibility

to support the authentication of messages. The manufacturer must set the default value to 4; it can be updated by the network administrator. The minimum security level must not be overridden by any devices and, as a consequence, the field `AllowedSecurityLevels` should contain only one value, equal to the minimum security level.

If the Hybrid Secured configuration is enabled, the minimum security level must be set to 0, thus supporting the joining of devices having different security capabilities. All the security levels could be allowed and the network administrator could decide to enable only a subset of them according to the network design.

If the Flexible Secured configuration is enabled, the minimum security level must be set to 1. The joining of nodes without (or with limited) security capabilities is permitted by setting the `DeviceOverrideSecurityMinimum` variable to `TRUE` and by including lower security levels in the list of `AllowedSecurityLevels`.

6 Initialization of a secured IEEE 802.15.4 domain

A secured 802.15.4 domain is created by means of three different phases: setting-up, bootstrapping, and key negotiation.

6.1 Setting-up phase

The setting-up phase is used to properly configure the device that will operate in an IEEE 802.15.4 network.

It consists in storing, within the device, parameters and initial secrets (i.e., the `masterKey`), which will be used by secure algorithms and procedure to setup the secure domain. They include. (i) the `MasterKey`, (ii) the `PrimeNumbersTable`, and (iii) the `GlobalSecurityLevelsTable`.

This operation may be performed by the manufacturer or by the network administrator. The `MasterKey` can be used to generate two different keys:

- the `DefaultKey`, exploited to protect broadcast messages (i.e., the beacon frame);
- the `LinkKey`, used to encrypt and authenticate unicast packets (i.e., those exchanged between only two specific nodes).

The `GlobalSecurityLevelsTable`, that has been reported in Fig. 3, is used to store the minimum security level and the list of allowed security levels that must be adopted for each kind of MAC frame and for each

security configuration defined in Sec. 5. The table reported in Fig. 3 does not consider ACK messages because they must not be protected (as imposed by the IEEE 802.15.4 standard [IEEE802154]).

Both the minimum security level and the list of allowed security levels must be chosen by the manufacturer or by the network administrator, according to restrictions reported in Fig. 2.

Attribute	Frame Type	Secured Network Configurations			
		Fully	Partial	Hybrid	Flexible
Security Minimum	Beacon				
Security Minimum	Data				
Security Minimum	Command MAC				
AllowedSecurityLevels	Beacon				
AllowedSecurityLevels	Data				
AllowedSecurityLevels	Command MAC				

Figure 3. Structure of the GlobalSecurityLevelsTable.

The PrimeNumbersTable stores a set of prime numbers and their respective primitive roots, which are used during the key negotiation procedure. Its implementation is reported in Fig. 4.

PrimeNumberId	Prime Number	Primitive Root
1	p1	g1
2	p2	g2
:		:
N	pN	gN

+-----+-----+-----+
 Figure 4. Structure of the PimeNumbersTable.

6.2 Bootstrap phase for a FFD device

The PAN is initialized by a FFD node. The MAC entity of the FFD node starts scanning the channel (for discovering the presence of other active coordinators and identifying the portion of the spectrum which it could operate in) after the reception of the MLME-START.request primitive, generated by the so called Next Higher Layer. Then, it will answer with a MLME-START.confirm primitive reporting a SUCCESS status. From this moment on, the device behaves as the PAN coordinator and it is able to select the identification number for the PAN, i.e., the PAN_ID, and the short MAC address of the IEEE 802.15.4 network [IEEE802154]. In this phase, the FFD generates the DefaultKey, D_k, and updates MAC security attributes accordingly. The DefaultKey, D_k, will be used to encrypt/decrypt all the broadcast messages. To this end, the following tasks will be executed.

- a) The DefaultKey, D_k, is obtained from the MasterKey, M_k, by using a 128-bit hash function, H_128(.)

$$D_k = H_{128}(PAN_ID \parallel M_k).$$

- b) The FFD creates the KeyDescriptor associated to the DefaultKey, D_k, by following these steps:

- b.1) A new KeyIdLookupList data structure is created and stored within the KeyDescriptor element. A KeyIdLookupDescriptor is generated and stored into the KeyIdLookupList data structure. The KeyIdMode, the KeySource, and the KeyIndex variables of this KeyIdLookupDescriptor are set to 0x03, the MAC address of the device, and 1, respectively. Instead, DeviceAddrMode, DevicePANId, and DeviceAddress are not set due to the selected KeyIdMode (see Tab. 65 of the IEEE 802.15.4 standard for more details [IEEE802154]).
 - b.2) A KeyUsageList data structure is created and stored within the KeyDescriptor element. One KeyUsageDescriptor for each broadcast message is create and stored into the KeyUsageList data structure.
 - b.3) The DeviceDescriptorHandleList is left blank because the FFD does not yet know the list of devices that may use

this key.

b.4) The DefaultKey, D_k, is stored within the Key field.

c) The KeyDescriptor created in the previous step is added to the macKeyTable.

d) The macDefaultKeySource is set to the MAC address of the device.

6.3 Bootstrap phase for a RFD device in a Beacon-enabled network

To join the network, the RFD device should associate with the coordinator. The Next Higher Layer sends to the MAC entity the MLME-ASSOCIATE.request primitive, starting the association procedure.

In this phase, the FFD generates the DefaultKey, D_k, and updates MAC security attributes accordingly. The DefaultKey, D_k, will be used to encrypt/decrypt all the broadcast messages. To this end, the following tasks will be executed:

a) The RFD node receives a Beacon messages sent by the coordinator and extracts from it the PAN_ID, the MAC address of the coordinator and the FrameCounter of the received frame.

b) A new DeviceDescriptor element, associated to the coordinator (i.e., the FFD node that sent the Beacon message) is created and stored into the macDeviceTable. It is built considering these specifications (see Tab. 64 of the IEEE 802.15.4 standard [IEEE802154] for more details):

b.1) The PANId variable is associated to the PAN_ID value extracted from the Beacon message.

b.2) The ShortAddress is set to the MAC address of the coordinator whenever the short addressing mode is used. This parameter is set to 0xffffe if only the extended addressing mode is used. If its value is unknown, the ShortAddress parameter is set to 0xffff.

b.3) The ExtAddress is set to the IEEE MAC address of the coordinator.

b.4) The FrameCounter parameter is set to the FrameCounter value extracted from the Beacon message.

b.5) The Exempt boolean flag is set to the allowed value of the DeviceOverrideSecurityMinimum variable described in Fig. 2.

c) The DefaultKey, D_k, is obtained from the MasterKey, M_k, by using a 128-bit hash function, H_128(.):

$$D_k = H_{128}(PAN_ID \parallel M_k).$$

d) The RFD creates the keyDescriptor associated to the DefaultKey, D_k, by following these steps:

d.1) a new KeyIdLookupList data structure is created and stored within the KeyDescriptor element. A KeyIdLookupDescriptor is generated and stored into the KeyIdLookupList data structure. The KeyIdMode, the KeySource, and the KeyIndex variables of this KeyIdLookupDescriptor are set to 0x03, the MAC address of the coordinator that sent the Beacon frame, and 1, respectively. Instead, DeviceAddrMode, DevicePANId, and DeviceAddress are not set due to the selected KeyIdMode (see Tab. 65 of the IEEE 802.15.4 standard for more details [IEEE802154]).

d.2) A KeyUsageList data structure is created and stored within the KeyDescriptor element. One KeyUsageDescriptor for each broadcast message is created and stored into the KeyUsageList data structure.

d.3) The DeviceDescriptorHandleList is created and populated with the pointer to the DeviceDescriptor created at the point 2.

d.4) The DefaultKey, D_k, is stored within the Key field.

e) The KeyDescriptor created in the previous step is added to the macKeyTable.

f) The macDefaultKeySource is set to the MAC address of the coordinator.

6.4 Bootstrap phase for a RFD device in a not-Beacon-enabled network

In the case the not-beacon-enabled scheme is enabled, the RFD device must explicitly request its generation to the coordinator. The payload of the Beacon Request packet must be protected using an ephemeral key, phi_k, obtained from the MasterKey, M_k, and the source address of the

RFD node, srcMACAddress, as

```
phi_k = H_128(srcMACAddress | M_K).
```

The KeyIdMode of the Beacon Request packet is set to 00, thus enabling the coordinator to implicitly obtain the ephemeral key.

Once received the Beacon frame, the RFD node will execute all the steps described in Sec. 6.3.

6.5 Key negotiation phase Since resource-constrained devices are unable to perform complex algorithms and protocols, a simple key agreement protocol is adopted during the execution of the key negotiation phase.

A new command message, which is identified with a CommandFrameIdentifier set to 0xAA, is used for this purpose. It is composed by four different fields: KeyGenControlField, Rand, KeyMaterial, and AuthenticationField.

The structure of the new command MAC frame has been reported in Fig. 5. The structure of the KeyGenControlField, instead, are shown in Fig. 6. The introduction of these new fields respects the constraints imposed by the standard about the maximum packet size.

Octets: 2	0/2	0/S	0/16
KeyGen ControlFiled	Rand	Key Material	Authentication Filed

Figure 5. A new command MAC frame adopted during the key negotiation phase.

Bits: 2	2	1	1	10
Message Type	KeyGen Mode	Key Flag	Auth Flag	Key Size

Figure 6. KeyGenControlField of the new command MAC frame adopted during the key negotiation phase.

The KeyGenControlField (2 bytes long) stores details about the content of the message. It is composed by the following fields:

- the MessageType (2 bits long), which identifies the type of message exchanged during the procedure. It may assume these

values:

- MessageType=00 identifies a message storing key materials (i.e., DH parameters).
- MessageType=01 identifies a message storing key materials belonging to a different approach selected before by the remote node. This message can be generate only by the FFD in the case the key negotiation algorithm chosen by the RFD is not supported.
- MessageType=10 identifies final messages belonging to the key negotiation phase that are used to verify the mutual authentication of nodes.
- MessageType=11 is reserved for future upgrades.
- the KeyGenMode (2 bits long), which describes the algorithm adopted for key generation. In this Internet draft we describe a key negotiation procedure based on the DH algorithm, which is identified by the code 00. Other values, i.e., 01, 10 and 11, are reserved and can be used for future upgrades.
- the boolean KeyFlag (1 bit long), which is set to TRUE in the case the message delivers key materials or to FALSE otherwise.
- the boolean AuthFlag (1 bit long)), which is set to TRUE in the case the message delivers an authentication field or to FALSE otherwise.
- the KeySize (10 bits long), which indicates the size of the transported key material. Its value is set to 0 in the case the message does not contain any key materials.

The Rand field (0/2 bytes long) contains a random value used for generating the PreLinkKey, P_k, and for verifying the authenticity of the remote device. It is present only if MessageType is equal to 00 or 01.

The KeyMaterial field (0/S bytes long, where S is the size of the prime number) contains key materials, such as DH parameters. It is present only if MessageType is equal to 00 or 01.

AuthenticationField field (0/16 bytes long) is used to verify the authenticity of the remote device. It is present only if MessageType is equal to 10.

6.5.1 Key exchange mechanism based on DH

The key exchange mechanism based on the DH algorithm is reported in Fig. 7.

It is initialized by the RFD device that wants to establish a secure link with the remote FFD. The procedure assumes that both RFD and FFD devices store into the PrimeNumbersTable the same set of N prime numbers and their primitive roots, each one having size equal to S (see Sec. 6.1 for more details).

The number of bits needed to identify each prime number of the PrimeNumbersTable, i.e., P_bits, is equal to

$$P_bits = \log_2 (N).$$

The key exchange mechanism is provides the execution of these operations:

- a) The RFD identifies in the PrimeNumbersTable a prime number, P, and the corresponding primitive root, g, by extracting the latest P_bits bits from the output of the following hash function

$$H_{128}(PAN_ID \mid D_k).$$

- b) The RFD computes the private key and the public key, according to the DH algorithm. Hence, the private key, PVK_RFD, is extracted as a random number. Then, the public key, PBK_RFD, is created as:

$$PBK_RFD = g^{PVK_RFD} \bmod P$$

- c) The RFD extract another random number, RAND_1, that will be used for the mutual authentication.

- d) The RDF sends its public key, PBK_RFD, to the remote FFD through a specific MAC command frame, which is composed by the following fields (see Fig. 7): MessageType=00, KeyGenMode=01, KeyFalg=TRUE, AuthFlag=FALSE, KeySize=S, Rand=RAND_1, and KeyMaterial=PBK_RFD. This message is encrypted with the DefaultKey, D_k.

- e) Once received the aforementioned MAC command frame, the FFD node generates, in turn, its private and public key through the DH algorithm. Hence, it identifies in the PrimeNumbersTable a prime number, P, and the corresponding primitive root, g, by extracting the latest P_bits bits from the output of the following hash function

$$H_{128} (PAN_ID \parallel D_k).$$

Then, it generates a random variable, PVK_FFD , which is the private key and computes the public key, PBK_FFD , as in the following:

$$PBK_FFD = g^{PVK_FFD} \bmod P.$$

f) The FFD extract another random number, $RAND_2$, that will be used for the mutual authentication;

f) The RDF sends its public key, PBK_FFD , to the remote RFD through a specific MAC command frame, which is composed by the following fields (see Fig. 7): $MessageType=00$, $KeyGenMode=01$, $KeyFalg=TRUE$, $AuthFlag=FALSE$, $KeySize=S$, $Rand=RAND_2$, and $KeyMaterial=PBK_FFD$. This message is encrypted with the $DefaultKey$, D_k .

h) The RFD computes the $PreLinkKey$, P_k

$$P_k = PBK_FFD^{PVK_RDF} \bmod P.$$

i) The FFD computes the $PreLinkKey$, P_k ,

$$P_k = PBK_RFD^{PVK_FFD} \bmod P.$$

j) Both RFD and FFD devices computes the $LinkKey$ by using the procedure described in Sec. 6.5.2 and update MAC security attributes according to procedures described in Secs. 6.5.3 and 6.5.4.

k) The RFD node computes the authentication parameters, $AUTH_RFD$, through the 128-bit hash function, $H_{128}()$,

$$AUTH_RFD = H_{128}(P_k \parallel RAND_2 \parallel RAND_1).$$

Then, it sends to the coordinator a new MAC command message to demonstrate its authenticity. This message is composed by the following fields (see Fig. 7): $MessageType=10$, $KeyGenMode=01$, $KeyFalg=FALSE$, $AuthFlag=TRUE$, $KeySize=0$, $Rand=RAND_2$, and $AuthenticationField=AUTH_RFD$. This message is protected by using the $LinkKey$ computed before.

l) The FFD node computes the authentication parameters, $AUTH_FFD$, through the 128-bit hash function, $H_{128}()$

$$AUTH_FFD = H_{128}(P_k \parallel RAND_1 \parallel RAND_2).$$

Then, it sends to the RFD node a new MAC command message to demonstrate its authenticity. This message is composed by the following fields(see Fig. 7): MessageType=10, KeyGenMode=01, KeyFalg=FALSE, AuthFlag=TRUE, KeySize=0, Rand=RAND_2, and AuthenticationField=AUTH_FFD. This message is protected through the LinkKey computed before.

m) Both RFD and FFD verify the authenticity of the remote.

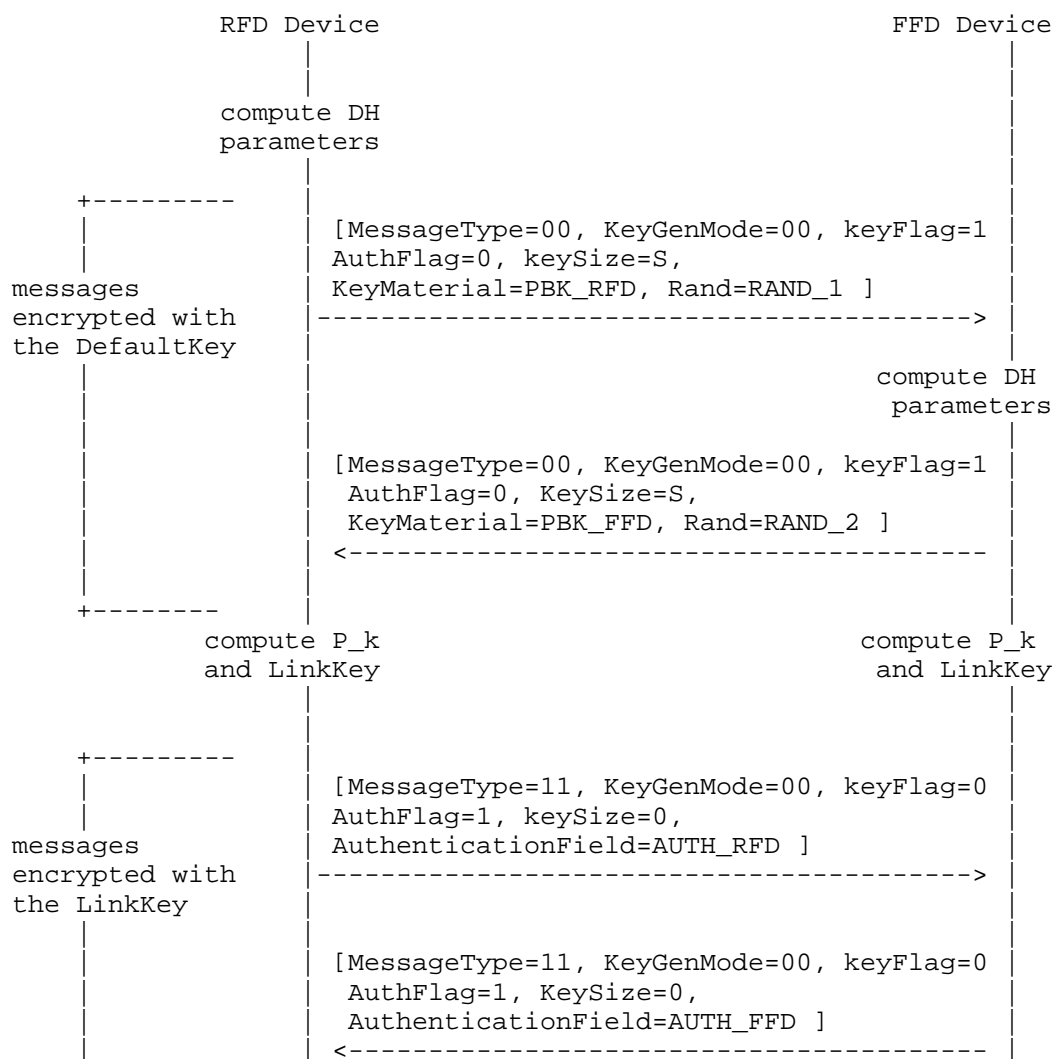




Figure 7. Key exchange mechanism based on DH.

6.5.2 Generation of the LinkKeys

The standard imposes to use the CCM* algorithm and a 128-bit key to protect MAC frames. Independently from the size the PreLinkKey, both RFD and FFD must create a set of link keys, each one 128 bits long. Firstly, each node computes a NewPreLinkKey, NP_k128, through the 128-bit hash function, H_128(.):

$$NP_k128 = H_128(PAN_ID \mid P_k).$$

The NP_k128 key will be used to compute LinkKeys. The CCM* algorithm assumes that each key must be used for a specific number of block ciphers. For each i-th group of block ciphers, the LinkKey, L_k, is computed as in the following:

$$L_k = H_128(i \mid PAN_ID \mid P_k).$$

Finally, both FFD and RFD devices update their MAC security attributes by using the procedures described in Secs. 6.5.3 and 6.5.4, respectively.

6.5.3 Update of MAC security attribute for the FFD node after the generation of the LinkKey

After the calculation of the i-th LinkKey, the FFD updates its MAC security attributes as described in what follows.

a) If $i=1$, a new DeviceDescriptor element, associated to the RFD node with which it has negotiated a link key, is created and stored into the macDeviceTable. It is composed by:

- a.1) the PANId, which is set to the PAN_ID value.
- a.2) The ShortAddress, which is set to the MAC address of

the RFD node whenever the short addressing mode is used. This parameter is set to 0xffffe if only the extended addressing mode is used. In the case its value is unknown, this parameter is set to 0xffff.

a.3) The ExtAddress, which is set to the IEEE MAC address of the RFD node.

a.4) The FrameCounter, which is set to the FrameCounter value extracted from the latest packet received by the RFD node.

a.5) The Exempt boolean flag, which is set to the allowed value of the DeviceOverrideSecurityMinimum variable described in Fig. 2.

b) The FFD creates the keyDescriptor associated to the i-th LinkKey, L_k, by following these steps:

b.1) A new KeyIdLookupList data structure is created and stored within the KeyDescriptor element. A KeyIdLookupDescriptor is generated and stored into the KeyIdLookupList data structure. The KeyIdMode, the KeySource, and the KeyIndex variables of this KeyIdLookupDescriptor are set to 0x03, the MAC address of the RFD node that initialized the key negotiation phase, and 1, respectively. DeviceAddrMode, DevicePANId, and DeviceAddress are not set because of the selected KeyIdMode (see Tab. 65 of the IEEE 802.15.4 standard for more details [IEEE802154]).

b.2) A KeyUsageList data structure is created and stored within the KeyDescriptor element. One KeyUsageDescriptor associated to data MAC frames is created and stored into the KeyUsageList data structure.

b.3) The DeviceDescriptorHandleList is created and populated with the pointer to the DeviceDescriptor created before.

b.4) The i-th portion of the LinkKey, L_k, is stored within the Key field.

c) The KeyDescriptor created in the previous step is added to the macKeyTable.

6.5.4 Update of MAC security attribute for the RFD node after the generation of the LinkKey

After the calculation of the i-th LinkKey, the RFD updates its MAC security attributes as described in what follows:

a) A keyDescriptor associated to the i-th LinkKey, L_k, is created by following these steps:

a.1) a new KeyIdLookupList data structure is created and stored within the KeyDescriptor element. A KeyIdLookupDescriptor is generated and stored into the KeyIdLookupList data structure. The KeyIdMode, the KeySource, and the KeyIndex variables of this KeyIdLookupDescriptor are set to 0x03, the MAC address of the RFD node, and 1, respectively. DeviceAddrMode, DevicePANId, and DeviceAddress are not set because of the selected KeyIdMode (see Tab. 65 of the IEEE 802.15.4 standard for more details [IEEE802154]).

a.2) A KeyUsageList data structure is created and stored within the KeyDescriptor element. One KeyUsageDescriptor associated to data MAC frames is created and stored into the KeyUsageList data structure.

a.3) The DeviceDescriptorHandleList is created and populated with the pointer to the DeviceDescriptor associate dto the coordinator.

a.4) The i-th portion of the LinkKey, L_k, is stored within the Key field.

b) The KeyDescriptor created in the previous step is added to the macKeyTable.

7 Additional features

There is the possibility to switch from the Flexible Secured to the Hybrid Secure configuration.

To this aim, during the association phase, a device without security capabilities sends to the coordinator a Beacon Request message with the SecurityEnabled flag set to FALSE.

The FFD device then switches to the Hybrid Secure configuration and update all the MAC security attributes accordingly.

From this moment on, the coordinator sends control messages in clear.

8 Security Considerations

There are no security considerations for this document.

9 IANA Considerations

There is no IANA action required for this document.

10 References

10.1 Normative References

- [KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC1776] Crocker, S., "The Address is the Message", RFC 1776, April 1 1995.
- [TRUTHS] Callon, R., "The Twelve Networking Truths", RFC 1925, April 1 1996.

10.2 Informative References

- [EVILBIT] Bellovin, S., "The Security Flag in the IPv4 Header", RFC 3514, April 1 2003.
- [RFC5513] Farrel, A., "IANA Considerations for Three Letter Acronyms", RFC 5513, April 1 2009.
- [RFC5514] Vyncke, E., "IPv6 over Social Networks", RFC 5514, April 1 2009.

Authors' Addresses

G. Piro
DEI, Dep. of Electrical and Information Engineering
Politecnico di Bari
Via Orabona 4, 70125, Bari, ITALY
Phone: +39 0805963301

Email: g.piro@poliba.it

G. Boggia
DEI, Dep. of Electrical and Information Engineering
Politecnico di Bari
Via Orabona 4, 70125, Bari, ITALY
Phone: +39 0805963913

Email: g.boggia@poliba.it

L.A. Grieco
DEI, Dep. of Electrical and Information Engineering
Politecnico di Bari
Via Orabona 4, 70125, Bari, ITALY
Phone: +39 0805963911

Email: a.grieco@poliba.it

6TiSCH
Internet-Draft
Intended status: Standards Track
Expires: April 24, 2014

R. Sudhaakar, Ed.
Cisco
P. Zand
University of Twente
October 21, 2013

6TiSCH Data Model for CoAP
draft-sudhaakar-6tisch-coap-00

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on April 24, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Requirements notation	2
2. Introduction	3
3. Scope of the document	3
4. Generic Data Model	4
5. Data Model definition for CoAP	4
5.1. Naming Convention for URI schemes	4
5.2. Convention for accessing URIs	4
5.3. 6TiSCH Resources	5
5.3.1. Management Resources	5
5.3.2. Informational Resources	7
5.3.3. Message Formats	7
5.3.4. Extensible Resources	10
5.4. Example	10
5.4.1. Request-Response	10
5.4.2. Publish-Subscribe	12
6. References	12
6.1. Normative References	12
6.2. Informative References	12
6.3. External Informative References	13
Appendix A.	13
Authors' Addresses	14

1. Requirements notation

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

2. Introduction

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

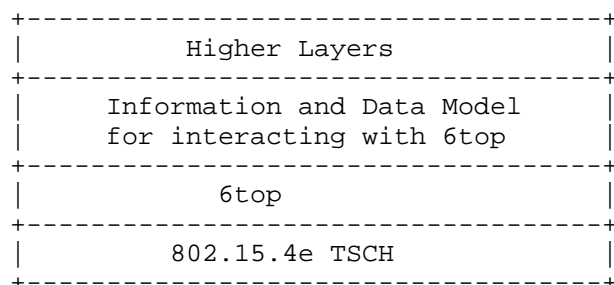


Figure 1: Logical positioning of layers

In order to have an wide impact we need to be able to interoperate with any protocol that may be used by the network layer. This documents aims at defining the message exchanges and the formats of the messages that the network layer uses to interact with the 6top sub-layer. We use the YANG data modelling language to specify a data format reusable across different protocols/elements including RPL, RSVP, PCE, etc.

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

3. Scope of the document

We first define a generic data model that is applicable to extensions on other transport protocols. The generic data model uses YANG to describe the message and formats that are used by the higher layers to interact with 6top.

It is followed by the implementation details of the data model for the specific scenario where the higher layer may use CoAP to interact with the 6top nodes. The document defines the URIs that are used to identify resources exposed by 6top. The messages that are required to be sent to the 6top sublayer are defined using the CBOR format.

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

4. Generic Data Model

[TODO]

5. Data Model definition for CoAP

5.1. Naming Convention for URI schemes

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

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

1. All URIs naming 6top resources MUST use the 'coap' scheme
2. The authority MUST have the username '6top' and the IP address of 6top node
3. The root path MUST always start with '6t'
4. Each component of the path SHOULD be of minimum possible length while being self descriptive.
5. Typographical conventions as described in A SHOULD be followed

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

5.2. Convention for accessing URIs

We use the GET, POST and DELETE methods described by CoAP. These methods MUST be used in accordance with their definition in Sec. 5.8 of [I-D.ietf-core-coap]. We have no need for the PUT method as the functionality of the POST method can be used for all situations that need updating or modification of a resource. The CoAP methods are mapped to 6top commands as shown in the figure below.

CoAP method	6top command	Description
-------------	--------------	-------------

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

Figure 2: Mapping between CoAP methods and 6top commands

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

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

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

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

5.3. 6TiSCH Resources

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

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

5.3.1. Management Resources

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

Name	Accessibility	URI path
------	---------------	----------

	6top Commands	
Neighbor Table	CREATE/READ/ DELETE/UPDATE	6t/Neighbor
slotframe Table	CREATE/READ/ DELETE/UPDATE	6t/slotframe
Cell Table	CREATE/READ/ DELETE/UPDATE	6t/Cell
Time Source	CREATE/READ/ DELETE/UPDATE	6t/TimeSource
Bundle Table	CREATE/READ/ DELETE/UPDATE	6t/Bundle
Track Table	CREATE/READ/ DELETE/UPDATE	6t/Track
EB Table	CREATE/READ/ DELETE/UPDATE	6t/EB

Figure 3: List of Management Resources

In the following table, we provide an example about how Neighbor table attributes can be addressed.

Field name	URI path
Neighbor Short Addr	6t/Neighbor/ShortAddr
numTx	6t/Neighbor/numTx
numTxAck	6t/Neighbor/numTxAck
numRx	6t/Neighbor/numRx
Neighbor Long Addr	6t/Neighbor/LongAddr
ASN	6t/Neighbor/ASN
RPL rank	6t/Neighbor/RPLrank

Time Source Flag	6t/Neighbor/TSFlag
RSSI	6t/Neighbor/RSSI
LQI	6t/Neighbor/LQI

Figure 4: Neighbor Table

5.3.2. Informational Resources

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

Name	Accessibility 6top Commands	URI path
Queue	READ/CONFIGURE	6t/Queue
Queue stats	READ/CONFIGURE	6t/QueueStats
Monitoring status	READ/CONFIGURE	6t/MonitoringStatus
Statistics metrics	READ/CONFIGURE	6t/StatisticsMetrics

Figure 5: List of Informational Resources

5.3.3. Message Formats

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

Header	GET
Uri-Path	/6t/Neighbor
Options	Accept: application/cbor Uri-Query: ABNF(ShortAddr==0x1234)

Figure 6: Example GET message

Since this resource points to the entire neighbor table the response returns all the rows (the list of neighbors of that node) and all fields in each row (i.e. entry for a neighbor) of the table in CBOR format. A request with a Uri-Query option may be used to retrieve only specific rows in the table. The value of Uri-Query MUST be in the ABNF format as described in [RFC5234].

Resources that point to collection within a table, such as '/6t/Neighbor/ShortAddr', returns only the values in the ShortAddr column of the Neighbor table. The usage of the Uri-Query option has the same effect of filtering on the result.

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

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

Header	POST	
Uri-Path	/6t/Neighbor	
Payload	CBOR({ShortAddr: 0x1234})	

Figure 7: Example POST message

The POST method may not be used on resources that are collection within a table, such as '/6t/Neighbor/ShortAddr'.

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

Header	DELETE
Uri-Path	/6t/Neighbor
Options	Uri-Query: ABNF(ShortAddr==0x1234)

Figure 8: Example DELETE message

A DELETE message SHOULD always contain a Uri-Query option in order to clearly specify which row(s) within the table must be deleted. Ideally, the CoAP client SHOULD make one call per row that must be deleted. An implementation may decide whether or not a DELETE method on '/6t/Neighbor' may be allowed.

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

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

CoAP method	6top command	6top behaviour	CoAP Response
POST /6t/Neighbor CBOR({ShortAddr: 1234})	Create.neighbor (address,stats)	Adds a neighbor	2.01 Created
GET /6t/Neighbor	Read.all. neighbor()	Reads all neighbors	2.05 Content CBOR(Neigh- bor Table)
GET /6t/Neighbor Uri-Query - ShortAddr == 0x1234	Read.neighbor (address)	Reads neighbor information	2.05 Content CBOR(Neigh- bor Table)
POST /6t/Neighbor CBOR({ShortAddr: 1234})	Update.neighbor (address,stats)	Updates an entry	2.04 Changed
DELETE /6t/Neighbor Uri-Query - ShortAddr == 0x1234	Delete.neighbor (address)	Removes the neighbor	2.02 Deleted

Figure 9: CoAP methods and resulting invocation 6top commands

5.3.4. Extensible Resources

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

5.3.4.1. Defining new resources

An extensible resource path MUST always start with '/6t/custom' and follow the guideline for URI naming as described in 5.1. The event associated with the extensible resource must be defined using the ABNF notation described in [RFC5234].

An extensible resource may be created by performing POST operation to the resource '/6t/custom' with the following payload encoded using CBOR.

Field Name	Type
Resource Name	String
Event Definition	String

Figure 10: Payload format for creating an Extensible Resource

5.3.4.2. Resource Profiles

[TODO]

5.3.4.3. Resource and Profile Discovery

[TODO]

5.4. Example

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

5.4.1. Request-Response

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

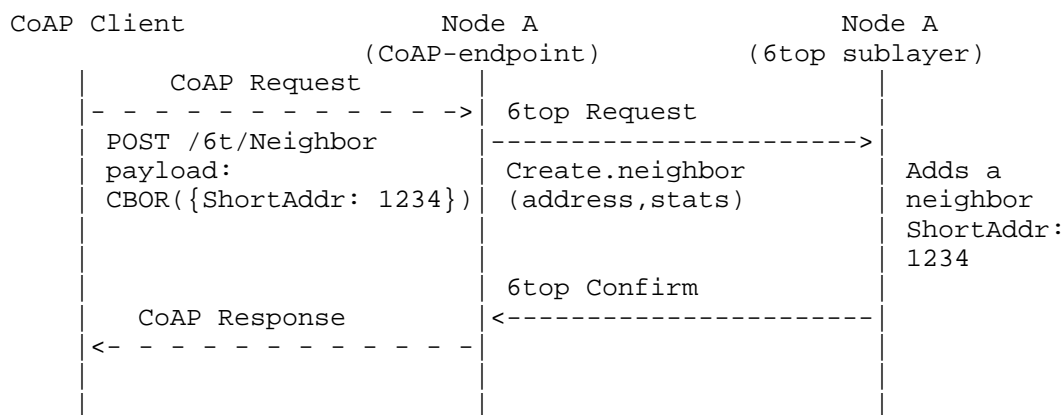


Figure 11: Example of adding a neighbor

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

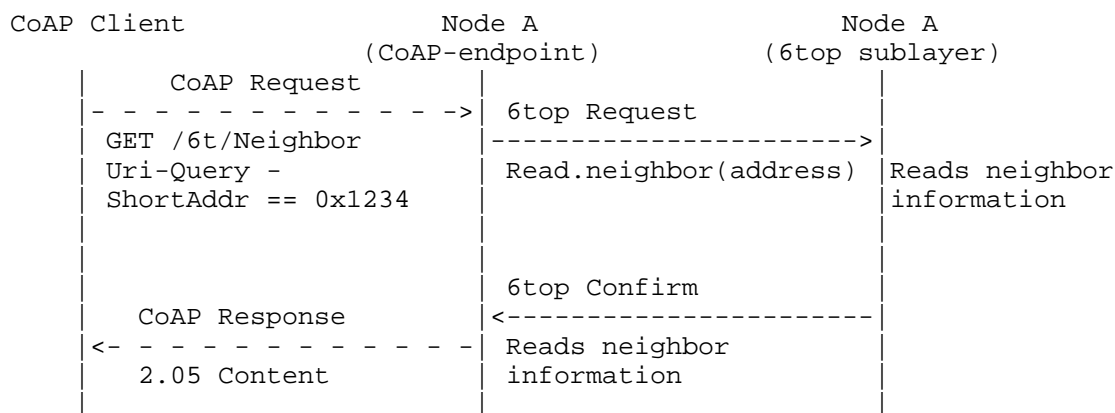


Figure 12: Example of reading a neighbor

5.4.2. Publish-Subscribe

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

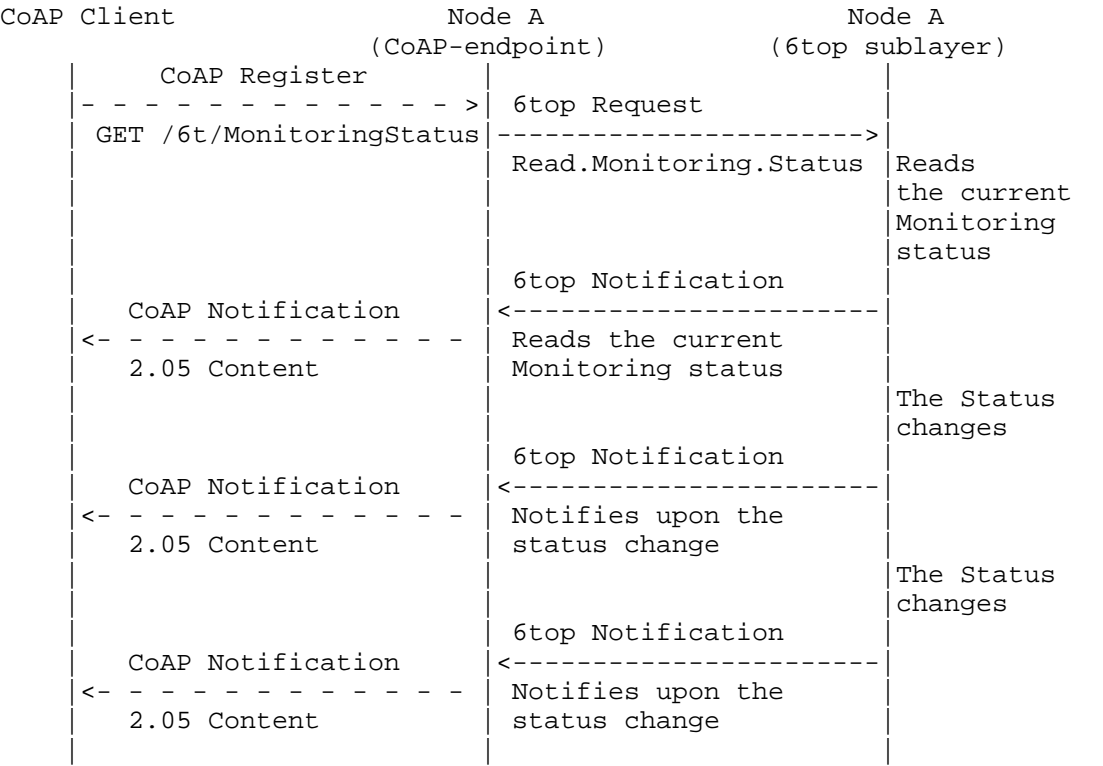


Figure 13: Example of Subscribing to Monitoring Status

6. References

6.1. Normative References

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

6.2. Informative References

[I-D.bormann-cbor]

Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", draft-bormann-cbor-09 (work in progress), September 2013.

[I-D.ietf-core-coap]

Shelby, Z., Hartke, K., and C. Bormann, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-18 (work in progress), June 2013.

[I-D.ietf-core-observe]

Hartke, K., "Observing Resources in CoAP", draft-ietf-core-observe-11 (work in progress), October 2013.

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

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

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

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

6.3. External Informative References

[IEEE802154e]

IEEE standard for Information Technology, "IEEE std. 802.15.4e, Part. 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer ", April 2012.

Appendix A.

Guidelines for constructing URI path names:

1. The first letter of each element of the path SHOULD be capitalized
2. If an element has multiple words, each the first letter of each word SHOULD be capitalized

Authors' Addresses

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

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

Pouria Zand
University of Twente
Graaf Florisstraat
1-F18
Deventer 7415 LK
Netherlands

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

6TiSCH
Internet-Draft
Intended status: Standards Track
Expires: April 22, 2014

P. Thubert, Ed.
Cisco
T. Watteyne
Linear Technology
RA. Assimiti
Centero
October 21, 2013

An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4e
draft-thubert-6tisch-architecture-01

Abstract

This document presents an architecture for an IPv6 Multi-Link subnet that is composed of a high speed powered backbone and a number of IEEE802.15.4e TSCH wireless networks attached and synchronized by Backbone Routers. Route Computation may be achieved in a centralized fashion by a Path Computation Element, in a distributed fashion using the Routing Protocol for Low Power and Lossy Networks, or in a mixed mode. The Backbone Routers perform proxy Neighbor Discovery operations over the backbone on behalf of the wireless device, so they can share a same subnet and appear to be connected to the same backbone as classical devices.

Requirements Language

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

Status of this Memo

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

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

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

This Internet-Draft will expire on April 22, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Terminology	3
3. Applications and Goals	4
4. Overview and Scope	4
5. Communication Paradigms and Interaction Models	7
6. Forwarding Models	8
6.1. Track Forwarding	8
6.1.1. Transport Mode	9
6.1.2. Tunnel Mode	9
6.1.3. Tunnel Metadata	10
6.2. Fragment Forwarding	11
6.3. IPv6 Forwarding	12
7. TSCH and 6top	12
7.1. 6top	12
7.2. Network Synchronization	13
7.3. Slotframes and Priorities	14
7.4. Packet Marking and Handling	14
8. Schedule Management Mechanisms	14
8.1. Minimal Static Scheduling	14
8.2. Neighbor-to-Nighbor Scheduling	15
8.3. Remote Monitoring and Schedule Management	15
8.4. Hop-by-hop Scheduling	16
9. Centralized vs. Distributed Routing	16
10. IANA Considerations	17
11. Security Considerations	17
12. Acknowledgements	17
13. References	17
13.1. Normative References	17
13.2. Informative References	18
13.3. External Informative References	19
Authors' Addresses	19

1. Introduction

The emergence of radio technology enabled a large variety of new types of devices to be interconnected, at a very low marginal cost compared to wire, at any range from Near Field to interplanetary distances, and in circumstances where wiring would be less than

practical, for instance rotating devices.

At the same time, a new breed of Time Sensitive Networks is being developed to enable traffic that is highly sensitive to jitter and quite sensitive to latency. Such traffic is not limited to voice and video, but also includes command and control operations such as found in industrial automation or in-vehicle sensors and actuators.

At IEEE802.1, the "Audio/Video Task Group", was renamed TSN for Time Sensitive Networking to address Deterministic Ethernet. The IEEE802.15.4 Medium access Control (MAC) has evolved with IEEE802.15.4e that provides in particular the Timeslotted Channel Hopping (TSCH) mode for industrial-type applications.

Though at a different time scale, both standards provide Deterministic capabilities to the point that a packet that pertains to a certain flow crosses the network from node to node following a very precise schedule, as a train that leaves intermediate stations at precise times along its path. With TSCH, time is formatted into timeslots, and an individual timeslot is allocated to unicast or broadcast communication at the MAC level. The time slotted operation reduces collisions, saves energy, and enables to more closely engineer the network for deterministic properties. The channel hopping aspect is a simple and efficient technique to combat multipath fading and external interference (for example by WiFi emitters).

This document presents an architecture for an IPv6 Multi-Link subnet that is composed of a high speed powered backbone and a number of IEEE802.15.4e TSCH wireless networks attached and synchronized by backbone routers. Route Computation may be achieved in a centralized fashion by a Path Computation Element (PCE), in a distributed fashion using the Routing Protocol for Low Power and Lossy Networks (RPL), or in a mixed mode. The Backbone Routers perform proxy IPv6 Neighbor Discovery (ND) operations over the backbone on behalf of the wireless devices, so they can share a same IPv6 subnet and appear to be connected to the same backbone as classical devices. Timeslots and other device resources are managed by an abstract Network Management Entity (NME) that may cooperate with the PCE in order to minimize the interaction with and the load on the constrained device.

2. Terminology

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

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

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

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

3. Applications and Goals

The architecture derives from existing industrial standards for Process Control by its focus on Deterministic Networking, in particular with the use of the IEEE802.15.4e TSCH MAC [IEEE802154e] and the centralized PCE. This approach leverages the TSCH MAC benefits for high reliability against interference, low-power consumption on deterministic traffic, and its Traffic Engineering capabilities. Deterministic Networking applies in particular to open and closed control loops, as well as supervisory control flows and management.

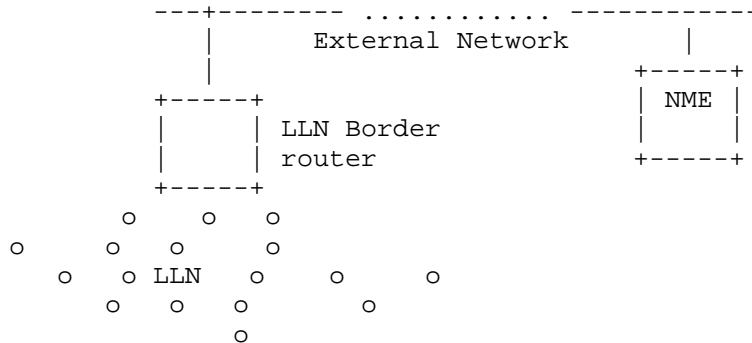
An incremental set of industrial requirements are addressed with the addition of an autonomic and distributed routing operation based on RPL. These use cases include plant setup and decommissioning, as well as monitoring of lots of lesser importance measurements such as corrosion and events. RPL also enables mobile use cases such as mobile workers and cranes.

A Backbone Router is included in order to scale the factory plant subnet to address large deployments, with proxy ND and time synchronization over a high speed backbone.

The architecture also applies to building automation that leverage RPL's storing mode to address multipath over a large number of hops, in-vehicle command and control that can be as demanding as industrial applications, commercial automation and asset Tracking with mobile scenarios, home automation and domotics which become more reliable and thus provide a better user experience, and resource management (energy, water, etc.).

4. Overview and Scope

The scope of the present work is a subnet that, in its basic configuration, is made of a IEEE802.15.4e Timeslotted Channel Hopping (TSCH) [I-D.watteyne-6tisch-tsch] MAC Low Power Lossy Network (LLN).



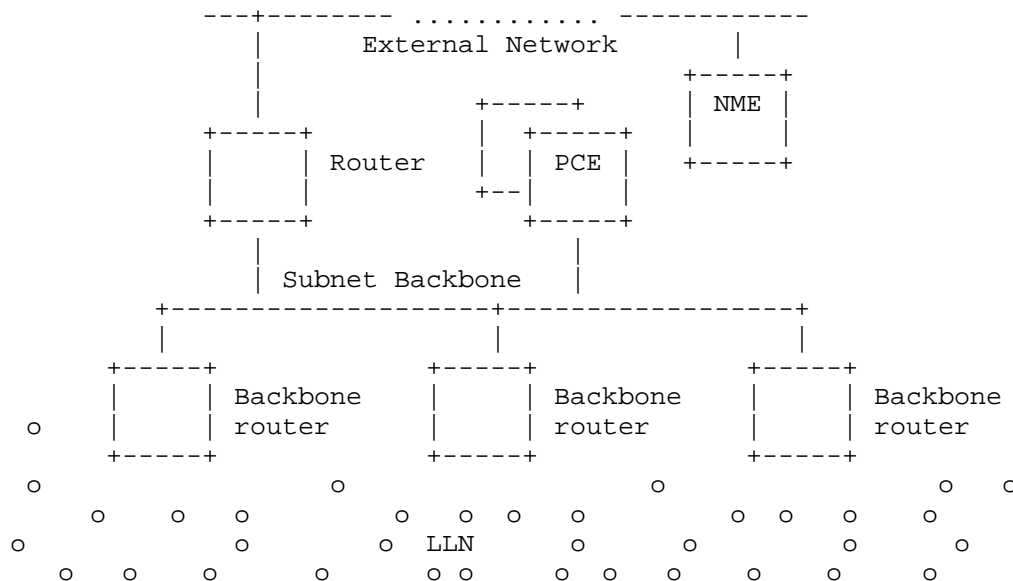
The LLN devices communicate over IPv6 [RFC2460] using the 6LoWPAN Header Compression (6LoWPAN HC) [RFC6282]. From the perspective of Layer 3, a single LLN interface (typically an IEEE802.15.4-compliant radio) may be seen as a collection of Links with different capabilities for unicast or multicast services. An IPv6 subnet spans over multiple links, effectively forming a Multi-Link subnet. Within that subnet, Neighbor Devices are discovered with 6LoWPAN Neighbor Discovery (6LoWPAN ND) [RFC6775]. The Routing Protocol for Low Power and Lossy Networks (RPL) [RFC6550] enables routing within the LLN, typically within the Multi-Link subnet in the so called Route Over fashion. RPL forms Destination Oriented Directed Acyclic Graphs (DODAGs) within Instances of the protocol, each Instance being associated with an Objective Function (OF) to form a routing topology. A particular LLN device, the LLN Border Router (LBR), acts as RPL root, 6LoWPAN HC terminator, and LLN Border Router (LBR) to the outside. The LBR is usually powered. More on RPL Instances can be found in [RFC6550], sections "3.1.2. RPL Identifiers" and "3.1.3. Instances, DODAGs, and DODAG Versions".

An extended configuration of the subnet comprises multiple LLNs. The LLNs are interconnected and synchronized over a backbone, that can be wired or wireless. The backbone can be a classical IPv6 network, with Neighbor Discovery operating as defined in [RFC4861] and [RFC4862]. The backbone can also support Efficiency-aware IPv6 Neighbor Discovery Optimizations [I-D.chakrabarti-nordmark-6man-efficient-nd] in mixed mode as described in [I-D.thubert-6lowpan-backbone-router].

Security is often handled at layer 2 and Layer 4. Authentication during the join process can be handled by the Protocol for Carrying Authentication for Network access (PANA) [RFC5191].

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

In the extended configuration, the functionality of the LBR is enhanced to that of Backbone Router (BBR). A BBR is an LBR, but also an Energy Aware Default Router (NEAR) as defined in [I-D.chakrabarti-nordmark-6man-efficient-nd]. The BBR performs ND proxy operations between the registered devices and the classical ND devices that are located over the backbone. 6TiSCH BBRs synchronize with one another over the backbone, so as to ensure that the multiple LLNs that form the IPv6 subnet stay tightly synchronized. If the Backbone is Deterministic (such as defined by the Time Sensitive Networking WG at IEEE), then the Backbone Router ensures that the end-to-end deterministic behavior is maintained between the LLN and the backbone.



The main architectural blocks are arranged as follows:

PCEP	CoAP	PANA	6LoWPAN	RPL
PCE	DTLS		ND	
TCP	UDP		ICMP	RSVP
IPv6				
6LoWPAN HC				
6top				
IEEE802.15.4e TSCH				

RPL is the routing protocol of choice for LLNs. (TBD RPL) whether there is a need to define a 6TiSCH OF.

(tbd NME) COMAN is working on network Management for LLN. They are considering the Open Mobile Alliance (OMA) Lightweight M2M (LWM2M) Object system. This standard includes DTLS, CoAP (core plus Block and Observe patterns), SenML and CoAP Resource Directory.

(tbd PCE) need to work with PCE WG to define flows to PCE, and define how to accommodate PCE routes and reservation. Will probably look a lot like GMPLS.

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

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

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

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

IoT6 European Project: external, maintain consistency.

5. Communication Paradigms and Interaction Models

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

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

would match a Data Model for the data that is provided over the protocol exchange.

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

This specification focusses on Communication Paradigms and Interaction Models for packet forwarding and TSCH resources (cells) management. Link-layer and Network-layer Packet forwarding interactions are discussed in Section 6, whereas Link-layer (one-hop), Network-layer (multithop along a track), and Application-layer (remote control) management mechanisms for the TSCH schedule are discussed in Section 8.

6. Forwarding Models

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

6.1. Track Forwarding

Track Forwarding is the simplest and fastest. A set of input cells are uniquely bound to a set of output cells, representing a forwarding state that can be used regardless of the upper layer protocol. This model can effectively be seen as a G-MPLS operation in that the information used to switch 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 Track is defined end-to-end as a succession of timeslots. A timeslot belongs to at most one Track. For a given iteration of a Slotframe, the timeslot is associated uniquely with a cell, which indicates the channel at which the timeslot operates for that iteration.

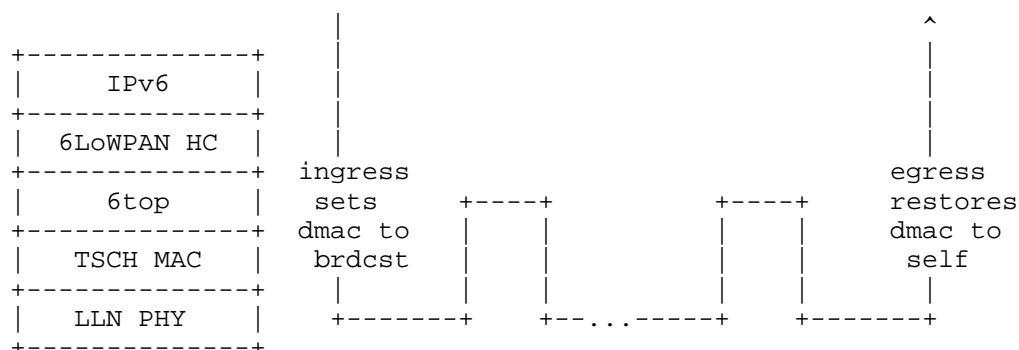
A data frame that is forwarded along a Track has a destination MAC address set to broadcast or a multicast address depending on MAC support. This way, the MAC layer in the intermediate nodes accepts the incoming frame and 6top switches it without incurring a change in the MAC header. In the case of IEEE802.15.4e, this means effectively broadcast, so that along the Track the short address for the destination is set to 0xFFFF.

Conversely, a frame that is received along a Track with a destination MAC address set to this node is extracted from the Track stream and delivered to the upper layer. A frame with an unrecognised MAC address is ignored at the MAC layer and thus is not received at the 6top sublayer.

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

6.1.1. Transport Mode

In transport mode, the PDU is associated flow information that refers uniquely to the Track, so the 6top sublayer can place the frame in the appropriate timeslot without ambiguity. In the case of IPv6 traffic, flow identification is transported in the Flow Label of the IPv6 header. Associated with the source IPv6 address, the flow label forms a globally unique identifier for that particular Track that is validated at egress before restoring the destination MAC address (dmac) and punting to the upper layer.

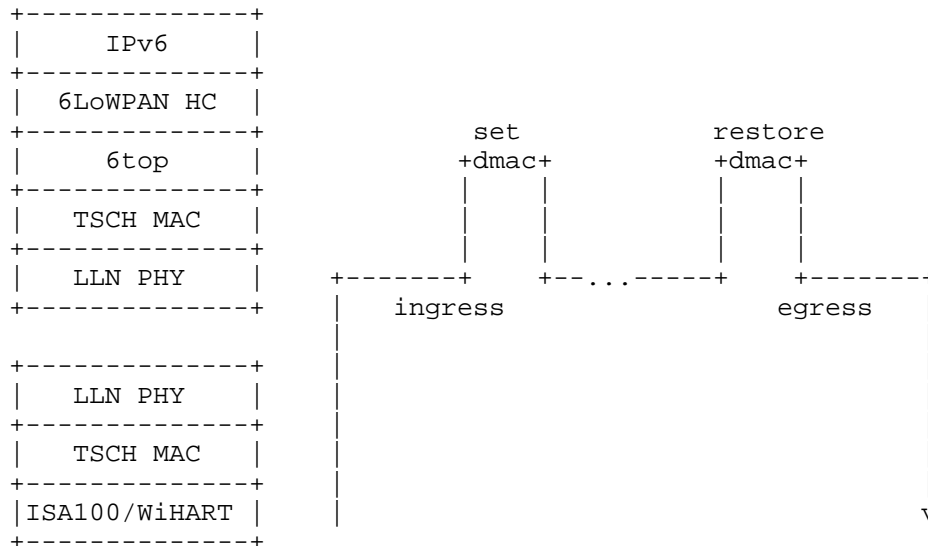


6.1.2. Tunnel Mode

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 to be transported transparently over the Track.



In that case, the flow information that identifies the Track is uniquely derived from the information at the receiving end, for instance the incoming timeslots, or an ISA100.11a ContractId. At the ingress 6TiSCH router, the packet destination is recognized as self but the flow information indicates that the frame must be tunnelled over a particular 6top Track so the packet is not punted to upper layer. Instead, it is passed to the 6top sublayer for switching. The 6top sublayer in the ingress router overrides the destination MAC to broadcast and forwards.

At the egress 6top router, the reverse operation occurs. Based on metadata associated to the Track, the frame is passed to the appropriate link layer with the destination MAC restored.

6.1.3. Tunnel Metadata

Metadata coming with the Track configuration is expected to provide the destination MAC address of the egress endpoint as well as the tunnel mode and specific data depending on the mode, for instance a service access point for frame delivery at egress. If the tunnel egress point does not have a MAC address that matches the configuration, the Track installation fails.

In transport mode, if the final layer 3 destination is the tunnel termination, then it is possible that the IPv6 address of the destination is compressed at the 6LoWPAN sublayer based on the MAC address. It is thus mandatory at the ingress point to validate that the MAC address that was used at the 6LoWPAN sublayer for compression matches that of the tunnel egress point. For that reason, the node that injects a packet on a Track checks that the destination is effectively that of the tunnel egress point before it overwrites it to broadcast. The 6top sublayer at the tunnel egress point reverts that operation to the MAC address obtained from the tunnel metadata.

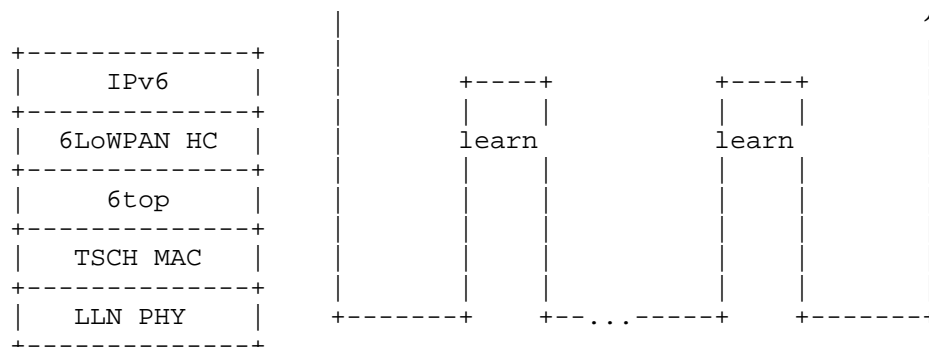
6.2. Fragment Forwarding

Considering that 6LoWPAN packets can be as large as 1280 bytes (the IPv6 MTU), and that the non-storing mode of RPL implies Source Routing that requires space for routing headers, and that a IEEE802.15.4 frame with security may carry in the order of 80 bytes of effective payload, an IPv6 packet might be fragmented into more than 16 fragments at the 6LoWPAN sublayer.

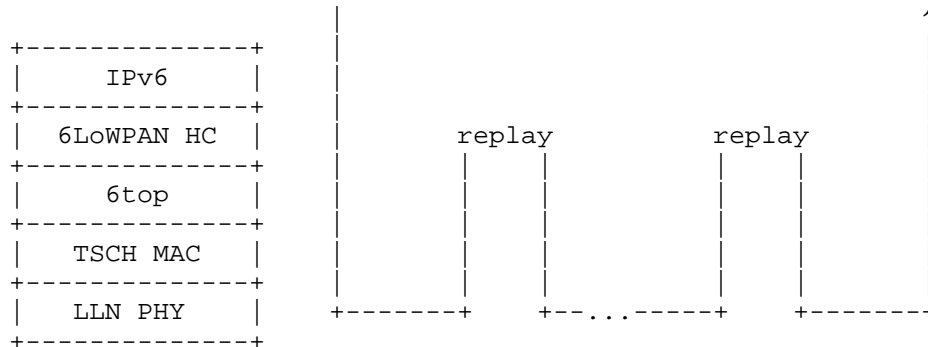
This level of fragmentation is much higher than that traditionally experienced over the Internet with IPv4 fragments, where fragmentation is already known as harmful.

In the case to a multihop route within a 6TiSCH network, Hop-by-Hop recomposition occurs at each hop in order to reform the packet and route it. This creates additional latency and forces intermediate nodes to store a portion of a packet for an undetermined time, thus impacting critical resources such as memory and battery.

[I-D.thubert-roll-forwarding-frags] describes a mechanism whereby the datagram tag in the 6LoWPAN Fragment is used as a label for switching at the 6LoWPAN sublayer. The draft allows for a degree of flow control base on an Explicit Congestion Notification, as well as end-to-end individual fragment recovery.



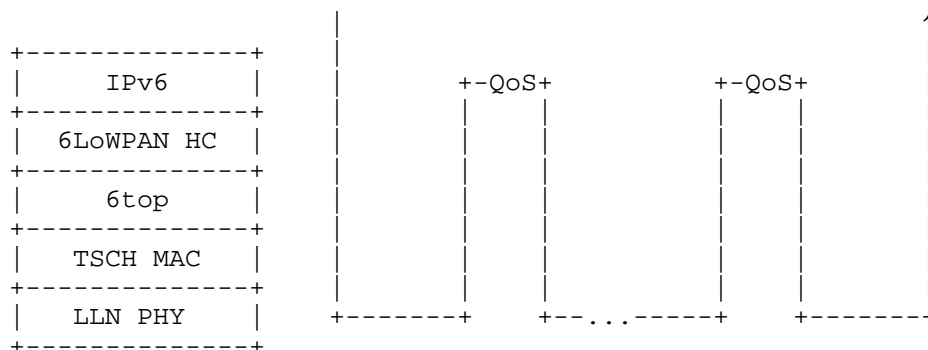
In that model, the first fragment is routed based on the IPv6 header that is present in that fragment. The 6LoWPAN sublayer learns the next hop selection, generates a new datagram tag for transmission to the next hop, and stores that information indexed by the incoming MAC address and datagram tag. The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



A bitmap and an ECN echo in the end-to-end acknowledgement enable the source to resend the missing fragments selectively. The first fragment may be resent to carve a new path in case of a path failure. The ECN echo set indicates that the number of outstanding fragments should be reduced.

6.3. IPv6 Forwarding

As the packets are routed at layer 3, traditional QoS and RED operations are expected to prioritize flows with differentiated services. A new class of service for Deterministic Forwarding is being defined to that effect in [I-D.svshah-tsvwg-lln-diffserv-recommendations].



7. TSCH and 6top

7.1. 6top

6top is a sublayer which is the next higher layer to TSCH and which offers a set of commands defining data and management interfaces. The management interface of 6top enables an upper layer to schedule cells and Slotframes in the TSCH schedule. 6top is defined in [I-D.wang-6tisch-6top].

If the scheduling entity explicitly specifies the slotOffset/channelOffset of the cells to be added/deleted, those cells are marked as "hard". 6top cannot move hard cells in the TSCH schedule. Hard cells are for example used by a central PCE.

6top contains a monitoring process which monitors the performance of cells, and can move a cell in the TSCH schedule when it performs bad. This is only applicable to cells which are marked as "soft". To reserve a soft cell, the higher layer does not indicate the exact slotOffset/channelOffset of the cell to add, but rather the resulting bandwidth and QoS requirements. When the monitoring process triggers a cell reallocation, the two neighbor nodes communicating over this cell negotiate its new position in the TSCH schedule.

7.2. Network Synchronization

Nodes in a TSCH network must be time synchronized. A node keeps synchronized to its time source neighbor through a combination of frame-based and acknowledgement-based synchronization. In order to maximize battery life and network throughput, it is advisable that RPL ICMP discovery and maintenance traffic (governed by the trickle timer) be somehow coordinated with the transmission of time synchronization packets (especially with enhanced beacons). This could be achieved through an interaction of the 6top sublayer and the RPL objective Function, or could be controlled by a management entity.

Time distribution requires a loop-less structure. Nodes taken in a synchronization loop will rapidly desynchronize from the network and become isolated. It is expected that a RPL DAG with a dedicated global Instance is deployed for the purpose of time synchronization. That Instance is referred to as the Time Synchronization Global Instance (TSGI). The TSGI can be operated in either of the 3 modes that are detailed in RPL [RFC6550] section "3.1.3. Instances, DODAGs, and DODAG Versions". Multiple uncoordinated DODAGs with independent roots may be used if all the roots share a common time source such as the Global Positioning System (GPS). In the absence of a common time source, the TSGI should form a single DODAG with a virtual root. A backbone network is then used to synchronize and coordinate RPL operations between the backbone routers that act as sinks for the LLN.

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

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

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

7.3. Slotframes and Priorities

6top uses priority queues to manage concurrent data flows of different priorities. When a packet is received from an higher layer for transmission, the I-MUX module of 6top inserts that packet in the outgoing queue which matches the packet best (DSCP can therefore be used). At each scheduled transmit slot, the MUX module looks for the frame in all the outgoing queues that best matches the cells. If a frame is found, it is given to TSCH for transmission.

7.4. Packet Marking and Handling

reservation Deterministic flow allocation (hard reservation of timeslots) eg centralized RSVP? metrics? Hop-by-hop interaction with 6top. Lazy reservation (use shared slots to transport extra burst and then dynamically (de)allocate) Classical QoS (dynamic based on observation)

8. Schedule Management Mechanisms

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

8.1. Minimal Static Scheduling

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

8.2. Neighbor-to-Neighbor Scheduling

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

Monitoring and relocation is done in the 6top layer. For the upper layer, the connection between two neighbor node appears as an number of cells. Depending on the 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.

8.3. Remote Monitoring and Schedule Management

[I-D.sudhaakar-6tisch-coap] defines an mapping of 6top's set of commands to CoAP resources. This allows an entity to interact with the 6top layer of a node that is multiple hops away. [I-D.sudhaakar-6tisch-coap] defines the CoAP resources and associated methods (GET/PUT/POST/DELETE). The payload of those signalling packets use CBOR to encode the different fields sent and received.

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

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

Query The CoAP client may retrieve information from a specific node in the network. This is typically a CoAP GET request issued on the appropriate resource on the node.

Report The CoAP client may register for periodic updates from a resource, for example to monitor the state of some statistics maintained by the node. This is typically done through CoAP Observe.

Action The CoAP client may request the node to take some action, for example add a cell to its TSCH schedule. This is typically a CoAP PUT/POST/DELETE request issued on the appropriate resource on the node.

Request The node may issue a request to the client to trigger some action, for example the calculation of a multi-hop route. This is typically a CoAP POST request issued by the node on the appropriate resource on the CoAP client.

Event The node may indicate the occurrence of a specific event to the CoAP client, for example the discovery of a new neighbor. This is typically a CoAP PUT request issued by the node on the appropriate resource on the CoAP client.

[I-D.sudhaakar-6tisch-coap] defines the a basic set of CoAP resources. For cases where extra functionality is needed, the draft also defines the concept of "profiles", as well as a mechanism for a CoAP client to discover the profiles installed on a node.

8.4. Hop-by-hop Scheduling

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

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

9. Centralized vs. Distributed Routing

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

Both may inject routes in the Routing Tables of the 6TiSCH routers. In either case, each route is associated with a topology that is indexed by an RPLInstanceID, as defined in RPL [RFC6550]. RPL and PCE rely on shared sources to define Global and Local RPLInstanceIDs.

It is possible for centralized and distributed routing to share a same topology. In this case, centralizes routes have precedence over distributed routes in case of a conflict.

Inside the 6TiSCH domain, the flow label is used to indicate the topology that must be used for routing. The associated Routing Tables are discussed in [I-D.thubert-roll-flow-label].

10. IANA Considerations

This specification does not require IANA action.

11. Security Considerations

This specification is not found to introduce new security threat.

12. Acknowledgements

13. References

13.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2460] Deering, S.E. and R.M. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", RFC 3444, January 2003.
- [RFC4080] Hancock, R., Karagiannis, G., Loughney, J. and S. Van den Bosch, "Next Steps in Signaling (NSIS): Framework", RFC 4080, June 2005.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W. and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC4862] Thomson, S., Narten, T. and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.
- [RFC5191] Forsberg, D., Ohba, Y., Patil, B., Tschofenig, H. and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA)", RFC 5191, May 2008.
- [RFC5889] Baccelli, E. and M. Townsley, "IP Addressing Model in Ad Hoc Networks", RFC 5889, September 2010.
- [RFC5974] Manner, J., Karagiannis, G. and A. McDonald, "NSIS Signaling Layer Protocol (NSLP) for Quality-of-Service Signaling", RFC 5974, October 2010.

- [RFC6282] Hui, J. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, September 2011.
- [RFC6550] Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP. and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, March 2012.
- [RFC6775] Shelby, Z., Chakrabarti, S., Nordmark, E. and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, November 2012.

13.2. Informative References

- [I-D.chakrabarti-nordmark-6man-efficient-nd]
Chakrabarti, S., Nordmark, E. and M. Wasserman,
"Efficiency aware IPv6 Neighbor Discovery Optimizations",
Internet-Draft draft-chakrabarti-nordmark-6man-efficient-nd-01, November 2012.
- [I-D.ietf-roll-rpl-industrial-applicability]
Phinney, T., Thubert, P. and R. Assimiti, "RPL
applicability in industrial networks", Internet-Draft
draft-ietf-roll-rpl-industrial-applicability-01, September
2013.
- [I-D.ohba-6tisch-security]
Chasko, S., Das, S., Lopez, R., Ohba, Y., Thubert, P. and
A. Yegin, "Security Framework and Key Management Protocol
Requirements for 6TiSCH", Internet-Draft draft-ohba-
6tisch-security-00, October 2013.
- [I-D.palattella-6tisch-terminology]
Palattella, M., Thubert, P., Watteyne, T. and Q. Wang,
"Terminology in IPv6 over the TSCH mode of IEEE
802.15.4e", Internet-Draft draft-palattella-6tisch-
terminology-00, October 2013.
- [I-D.sudhaakar-6tisch-coap]
Watteyne, T., "6TiSCH Data Model for CoAP", Internet-Draft
draft-sudhaakar-6tisch-coap-00, October 2013.
- [I-D.svshah-tsvwg-lln-diffserv-recommendations]
Shah, S. and P. Thubert, "Differentiated Service Class
Recommendations for LLN Traffic", Internet-Draft draft-
svshah-tsvwg-lln-diffserv-recommendations-00, February
2013.
- [I-D.thubert-6lowpan-backbone-router]
Thubert, P., "6LoWPAN Backbone Router", Internet-Draft
draft-thubert-6lowpan-backbone-router-03, February 2013.

- [I-D.thubert-roll-flow-label]
Thubert, P., "Use of the IPv6 Flow Label within an LLN",
Internet-Draft draft-thubert-roll-flow-label-02, November
2012.
- [I-D.thubert-roll-forwarding-frags]
Thubert, P. and J. Hui, "LLN Fragment Forwarding and
Recovery", Internet-Draft draft-thubert-roll-forwarding-
frags-01, February 2013.
- [I-D.vilajosana-6tisch-minimal]
Vilajosana, X. and K. Pister, "Minimal 6TiSCH
Configuration", Internet-Draft draft-vilajosana-6tisch-
minimal-00, October 2013.
- [I-D.wang-6tisch-6top]
Wang, Q., Vilajosana, X. and T. Watteyne, "6TiSCH
Operation Sublayer (6top)", Internet-Draft draft-wang-
6tisch-6top-00, October 2013.
- [I-D.watteyne-6tisch-tsch]
Watteyne, T., "Using IEEE802.15.4e TSCH in an LLN context:
Overview, Problem Statement and Goals", Internet-Draft
draft-watteyne-6tisch-tsch-00, October 2013.

13.3. External Informative References

- [HART] www.hartcomm.org, "Highway Addressable Remote Transducer,
a group of specifications for industrial process and
control devices administered by the HART Foundation", .
- [IEEE802.1TSNTG]
IEEE Standards Association, "IEEE 802.1 Time-Sensitive
Networks Task Group", March 2013, <[http://www.ieee802.org/
1/pages/avbridges.html](http://www.ieee802.org/1/pages/avbridges.html)>.
- [IEEE802154e]
IEEE standard for Information Technology, "IEEE std.
802.15.4e, Part. 15.4: Low-Rate Wireless Personal Area
Networks (LR-WPANs) Amendment 1: MAC sublayer", April
2012.
- [ISA100.11a]
ISA, "ISA100, Wireless Systems for Automation", May 2008,
<[http://www.isa.org/Community/
SP100WirelessSystemsforAutomation](http://www.isa.org/Community/SP100WirelessSystemsforAutomation)>.

Authors' Addresses

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

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

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

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

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

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

6TiSCH
Internet-Draft
Intended status: Informational
Expires: April 12, 2014

X. Vilajosana, Ed.
Universitat Oberta de Catalunya
K. Pister
University of California Berkeley
October 09, 2013

Minimal 6TiSCH Configuration
draft-vilajosana-6tisch-minimal-00

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on April 12, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Minimal Schedule Configuration	3
2.1. Slotframe	3
2.2. Cell Options	5
2.3. Retransmissions	6
2.4. Time Slot timing	6
3. Enhanced Beacons Configuration and Content	7
3.1. Sync IE	8
3.1.1. IE Header	8
3.1.2. IE Content	8
3.2. Frame and Cell IE	8
3.2.1. IE Header	9
3.2.2. IE Content	9
4. Acknowledgement	9
4.1. ACK/NACK Time Correction IE	9
4.1.1. IE Header	9
4.1.2. IE Content	10
5. Neighbour information	10
5.1. Neighbour Table	10
5.2. Time Source Neighbour Selection	11
6. Queues and Priorities	11
7. RPL on TSCH	12
7.1. RPL Objective Function Zero	12
7.1.1. Rank computation	12
7.1.2. Rank computation Example	13
7.2. RPL Configuration	15
7.2.1. Mode of Operation	15
7.2.2. Trickle Timer	16
7.2.3. Hysteresis	16
8. Acknowledgements	16
9. References	16
9.1. Normative References	16
9.2. Informative References	17
9.3. External Informative References	18
Authors' Addresses	18

1. Introduction

The nodes in a [IEEE802154e] TSCH network follow a communication schedule. The entity (centralized or decentralized) responsible for building and maintaining that schedule has very precise control over the trade-off between the network's latency, bandwidth, reliability and power consumption. During early interoperability testing and

development, however, simplicity is often more important than efficiency. One goal of this document is to define the simplest set of rules for building a [IEEE802154e] TSCH-compliant network, at the necessary price of lesser efficiency. Yet, this minimal mode of operation can also be used during network bootstrap before any schedule is installed into the network so nodes can self organize and the management and configuration information be distributed. In addition, as outlined in [I-D.phinney-roll-rpl-industrial-applicability] the minimal configuration can be used as a fallback mode of operation, ensuring connectivity of nodes in case that dynamic scheduling mechanisms fail or are not available. [IEEE802154e] provides a mechanism whereby the details of slotframe length, timeslot timing, and channel hopping pattern are communicated at synchronization to a node, also Enhanced Beacons can be used to periodically update nodes information. This document describes specific settings for these parameters. Nodes SHOULD broadcast properly formed Enhanced Beacons to announce these values, but during initial implementation and debugging it may be convenient to hard-code these values.

2. Minimal Schedule Configuration

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

2.1. Slotframe

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

Minimal configuration

Property	Value
Number of time slots per Slotframe	101
Number of available channels	16
Number of EBs cells	1 (slotOffset 0)
Number of scheduled cells	5 (slotOffsets 1,2,3,4,5)
Number of unscheduled cells	95 (from slotOffset 6 to 100)
Number of MAC retransmissions (max)	3
Time Slot duration	15ms

The suggested minimal schedule may be hard-coded in each node. The slotframe is composed of 101 time slots. The first slot in the slotframe is used to send Enhanced Beacons announcing the presence of the network. These EBs are not acknowledged. Five cells are scheduled for exchanging data packets, as described in Section 2.2. These cells are scheduled at slotOffset 1 to 5, and channelOffset 0. Per the IEEE802.15.4e TSCH, data packets sent on these cells to a unicast MAC address are acknowledged by the receiver. The 95 remaining cells are unscheduled, but are available to be allocated by dynamic scheduling solutions.

Minimal schedule overview



EB: Enhanced Beacon

Tx: Transmit

Rx: Receive

S: Shared

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

2.2. Cell Options

Per the [IEEE802154e] TSCH, each scheduled cell has a bitmap of cell options assigned, named LinkOption. All scheduled cells in the minimal schedule are configured as Hard cells [I-D.watteyne-6tsch-tsch-lln-context][I-D.wang-6tsch-6top]. Additional available cells can be scheduled by a dynamic scheduling solution and can either be configured as hard cells or soft cells without any restriction.

The EB cell is assigned the following bitmap of cell options:

b0 = Transmit = 1 (set)

b1 = Receive = 0 (clear)

b2 = Shared = 0 (clear)

b3 = Timekeeping = 0 (clear)

b4 = Hard = 1 (set)

b5-b7 = Reserved (clear)

The data cells are assigned the bitmap of cell options below that results in "Slotted Aloha" behaviour. Because both the "Transmit" and "Receive" bits are set, a node either transmits, if there is a packet in its queue, or listens if it has nothing to transmit. Because the "shared" bit is set, the back-off mechanism defined in [IEEE802154e] is used to resolve contention.

b0 = Transmit = 1 (set)

b1 = Receive = 1 (set)

b2 = Shared = 1 (set)

b3 = Timekeeping = 0 (clear)

b4 = Hard = 1 (set)

b5-b7 = Reserved (clear)

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

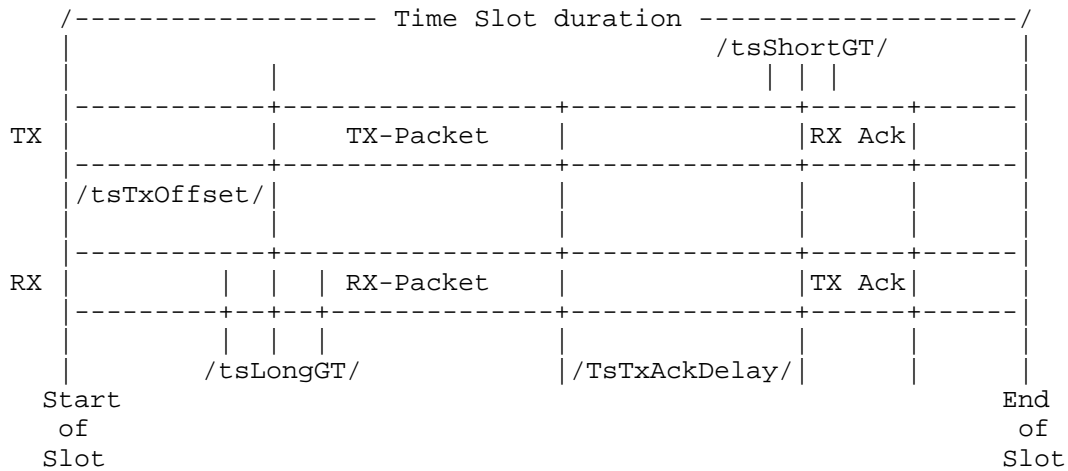
2.3. Retransmissions

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

2.4. Time Slot timing

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

Time slot internal timing diagram



[IEEE802154e] does not define the different durations of a time slot. It does allow those durations to be sent in the EBs (through a TimeSlot IE). This document recommends to pre-configure the different durations to the values listed below or use EBs to learn those values included in the TimeSlot IE.

Timeslot durations

IEEE802.15.4e TSCH parameter	Value
TsTxOffset	4000us
TsLongGT	2600us
TsTxAckDelay	4606us
TsShortGT	1000us
Time Slot duration	15000us

3. Enhanced Beacons Configuration and Content

[IEEE802154e] does not define how often or which EBs are sent. The choice of the duration between two EBs needs to take into account

whether EBs are used as the only mechanism to synchronize devices, or whether a Keep-Alive (KA) mechanism is used in parallel. For a simplest TSCH configuration, a mote SHOULD send an EB every 10s. For additional reference see [I-D.wattheyne-6tsch-tsch-lln-context] where different synchronization approaches are summarized.

EBs MUST be sent with the Beacon IEEE802.15.4 frame type and this EBs MUST carry the following Information Elements (IEs): (The content of the IEs is presented here for clarity, however this information is redundant with [I-D.wattheyne-6tsch-tsch-lln-context] and [IEEE802154e].)

3.1. Sync IE

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

3.1.1. IE Header

Length (b0-b7) = 0x06
Sub-ID (b8-b14) = 0x1a
Type (b15) = 0x00 (short)

3.1.2. IE Content

ASN Byte 1 (b16-b23)
ASN Byte 2 (b24-b31)
ASN Byte 3 (b32-b39)
ASN Byte 4 (b40-b47)
ASN Byte 5 (b48-b55)
Join Priority (b56-b63)

3.2. Frame and Cell IE

Although the schedule may be hard-coded during development, each node MUST indicate the schedule in each EB through a Frame and Cell IE. This enables nodes which implement [IEEE802154e] fully to configure their schedule as they join the network, and interact with nodes using a hard-coded schedule.

3.2.1. IE Header

Length (b0-b7) = variable

Sub-ID (b8-b14) = 0x1b

Type (b15) = 0x00 (short)

3.2.2. IE Content

Slotframes (b16-b23) = 0x01

Slotframe ID (b24-b31) = 0x01

Size Slotframe (b32-b47) = 0x65 (101)

Links (b48-b55) = 0x06

For each link in the minimal schedule:

Channel Offset (2B) = 0x00

Slot Number (2B) = from 0x00 to 0x05

LinkOption (1B) = as described in Section 2.2

4. Acknowledgement

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

4.1. ACK/NACK Time Correction IE

The ACK/NACK time correction IE is used to carry the measured de-synchronization between the sender and the receiver.

4.1.1. IE Header

Length (b0-b7) = 0x02

Sub-ID (b8-b14) = 0x1e

Type (b15) = 0x00 (short)

4.1.2. IE Content

Time Synch Info and ACK status (b16-b31)

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

ACK status and Time Synchronization information.

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

5. Neighbour information

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

5.1. Neighbour Table

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

Neighbour statistics:

numTx: number of transmitted packets to that neighbour

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

numRx: number of received packets from that neighbour

The EUI64 of the neighbour address.

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

RPL rank of that neighbour.

A flag which indicates whether this neighbour is a time source neighbour.

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

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

5.2. Time Source Neighbour Selection

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

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

6. Queues and Priorities

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

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

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

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

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

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

7. RPL on TSCH

Nodes in the network MUST use the RPL routing protocol

7.1. RPL Objective Function Zero

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

7.1.1. Rank computation

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

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

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

Where:

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

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

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

R_f (rank_factor): A configurable factor that is used to multiply the effect of the link properties in the rank_increase computation. If none is configured, then a rank_factor of 1 is used. For the purpose of this document rank_factor MUST be set to 1.

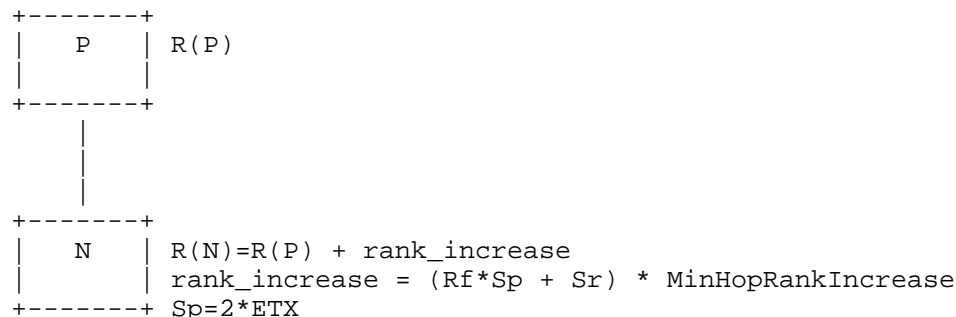
Sp (step_of_rank): (strictly positive integer) - an intermediate computation based on the link properties with a certain neighbour. For the purpose of this document $2*ETX$ (Expected Transmissions) as defined by [DeCouto03] and [RFC6551] MUST be used. The ETX will be computed as the inverse of the Packet Delivery Ratio (PDR) computed as the number of acknowledged packets divided by the number of transmitted packets to a certain node. E.g: $Sp=2*numTX/numTXAck$

Sr (stretch_of_rank): (unsigned integer) - the maximum augmentation to the step_of_rank of a preferred parent to allow the selection of an additional feasible successor. If none is configured to the device, then the step_of_rank is not stretched. For the present document stretch_of_rank MUST be set to 0.

MinHopRankIncrease: the MinHopRankIncrease is set to the fixed constant DEFAULT_MIN_HOP_RANK_INCREASE [RFC6550]. DEFAULT_MIN_HOP_RANK_INCREASE has a value of 256.

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

Rank computation scenario



7.1.2. Rank computation Example

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

$$R_f = 1$$

$$S_p = 2 * ETX$$

$$S_r = 0$$

$$\text{minHopRankIncrease} = 256 \text{ (default in RPL)}$$

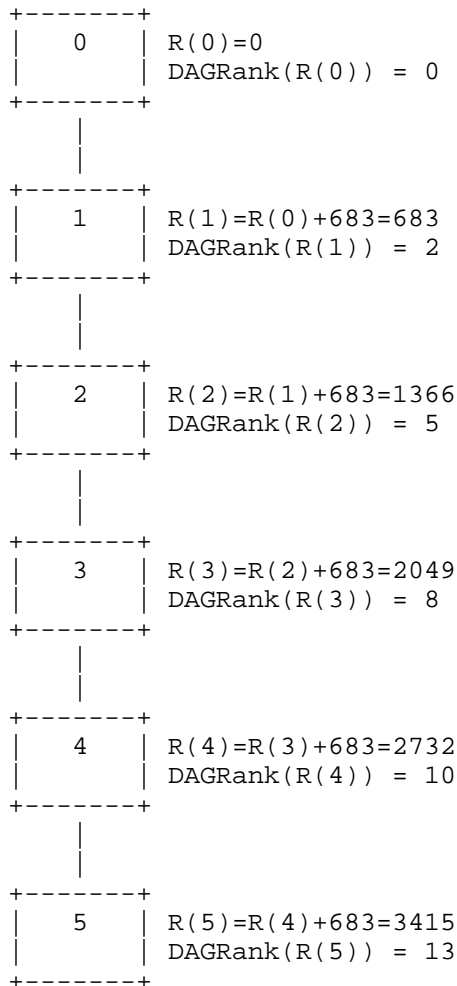
$$ETX = (\text{numTX} / \text{numTXAck})$$

$$r(n) = r(p) + \text{rank_increase}$$

$$\text{rank_increase} = (R_f * S_p + S_r) * \text{minHopRankIncrease}$$

$$\text{rank_increase} = 512 * \text{numTx} / \text{numTxACK}$$

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



7.2. RPL Configuration

In addition to the Objective Function (OF), a minimal configuration for RPL should indicate the preferred mode of operation and trickle timer operation so different RPL implementations can interoperate.

7.2.1. Mode of Operation

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

7.2.2. Trickle Timer

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

7.2.3. Hysteresis

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

8. Acknowledgements

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

9. References

9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC6206] Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko, "The Trickle Algorithm", RFC 6206, March 2011.
- [RFC6550] Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, March 2012.
- [RFC6551] Vasseur, JP., Kim, M., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks", RFC 6551, March 2012.

- [RFC6552] Thubert, P., "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6552, March 2012.
- [RFC6719] Gnawali, O. and P. Levis, "The Minimum Rank with Hysteresis Objective Function", RFC 6719, September 2012.

9.2. Informative References

- [I-D.wattheyne-6tsch-tsch-lln-context]
Wattheyne, T., Palattella, M., and L. Grieco, "Using IEEE802.15.4e TSCH in an LLN context: Overview, Problem Statement and Goals", draft-wattheyne-6tsch-tsch-lln-context-02 (work in progress), May 2013.
- [I-D.thubert-6tsch-architecture]
Thubert, P., Assimiti, R., and T. Wattheyne, "An Architecture for IPv6 over Timeslotted Channel Hopping", draft-thubert-6tsch-architecture-02 (work in progress), July 2013.
- [I-D.palattella-6tsch-terminology]
Palattella, M., Thubert, P., Wattheyne, T., and Q. Wang, "Terminology in IPv6 over Timeslotted Channel Hopping", draft-palattella-6tsch-terminology-01 (work in progress), July 2013.
- [I-D.wang-6tsch-6top]
Wang, Q., Vilajosana, X., and T. Wattheyne, "6TSCH Operation Sublayer (6top)", draft-wang-6tsch-6top-00 (work in progress), July 2013.
- [I-D.ohba-6tsch-security]
Chasko, S., Das, S., Lopez, R., Ohba, Y., Thubert, P., and A. Yegin, "Security Framework and Key Management Protocol Requirements for 6TSCH", draft-ohba-6tsch-security-01 (work in progress), July 2013.
- [I-D.ietf-roll-terminology]
Vasseur, J., "Terms used in Routing for Low power And Lossy Networks", draft-ietf-roll-terminology-13 (work in progress), October 2013.
- [I-D.phinney-roll-rpl-industrial-applicability]
Phinney, T., Thubert, P., and R. Assimiti, "RPL applicability in industrial networks", draft-phinney-roll-rpl-industrial-applicability-02 (work in progress), February 2013.

9.3. External Informative References

[IEEE802154e]

IEEE standard for Information Technology, "IEEE std. 802.15.4e, Part. 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANS) Amendment 1: MAC sublayer", April 2012.

[IEEE802154]

IEEE standard for Information Technology, "IEEE std. 802.15.4, Part. 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks", June 2011.

[DeCouto03]

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

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

Authors' Addresses

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

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

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

Email: pister@eecs.berkeley.edu

6TiSCH
Internet-Draft
Intended status: Informational
Expires: April 23, 2014

Q. Wang, Ed.
Univ. of Sci. and Tech. Beijing
X. Vilajosana
Universitat Oberta de Catalunya
T. Watteyne
Linear Technology
October 20, 2013

6TiSCH Operation Sublayer (6top)
draft-wang-6tisch-6top-00

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on April 23, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
2. 6TiSCH Operation Sublayer (6top)	4
2.1. Overview	4
2.2. Cell Model	5
2.2.1. hard cells	7
2.2.2. soft cells	7
2.3. Data Convey Model	7
2.4. Commands	9
2.4.1. Cell Commands	11
2.4.2. Slotframe Commands	14
2.4.3. Monitoring Commands	15
2.4.4. Statistics Commands	16
2.4.5. Network Formation Commands	17
2.4.6. Time Source Neighbor Commands	19
2.4.7. Neighbor Commands	20
2.4.8. Queueing Commands	21
2.4.9. Security Commands	23
2.4.10. Data Commands	25
2.4.11. Label Switching Commands	26
2.5. Message Formats	27
2.5.1. Information Elements	27
2.5.2. Packet Formats	35
2.6. Time Sequence	40
2.6.1. Network Formation	40
2.6.2. Creating soft cells	41
2.6.3. Deleting soft cells	42
2.6.4. Maintaining soft cells	42
2.6.5. Creating hard cells	43
2.6.6. Deleting hard cells	43
2.7. Statistics	43
2.7.1. Statistics Metrics	43

2.7.2. Statistics Configuration	44
2.8. Monitoring	44
2.8.1. Monitor Configuration	44
2.8.2. Actuation	45
2.9. Label Switching	45
3. Using 6top	46
3.1. RPL on 6top	46
3.1.1. Support to Neighbor Discovery and Parent Selection .	46
3.1.2. Support of Rank Computation	47
3.1.3. Support of Control Messages Broadcast	47
3.1.4. Support for QoS	48
3.2. GMPLS on 6top	49
3.2.1. Cell Reservation Support for GMPLS on 6top	50
3.2.2. Supporting QoS	50
4. References	50
4.1. Normative References	50
4.2. Informative References	51
4.3. External Informative References	54
Authors' Addresses	55

1. Introduction

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

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

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

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

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

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

2. 6TiSCH Operation Sublayer (6top)

2.1. Overview

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

Protocol Stack

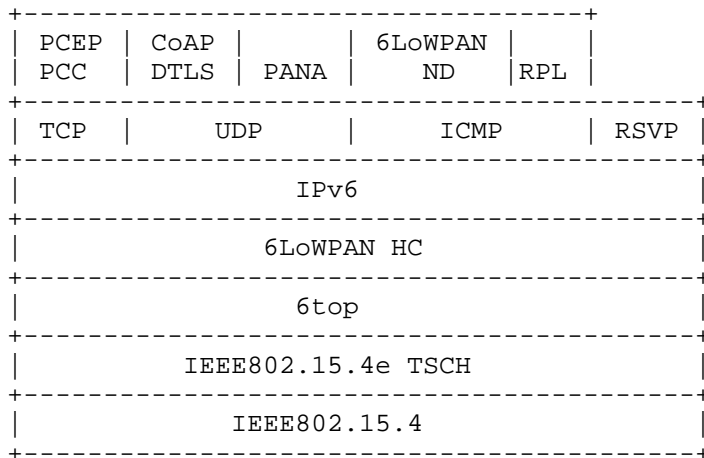


Figure 1

6top distinguishes between hard cells and soft cells. It therefore requires an extra flag to all cells in the TSCH schedule, as detailed in Section 2.2.

When a higher layer gives 6top a 6LoWPAN packet for transmission, 6top maps it to the appropriate outgoing priority-based queue, as detailed in Section 2.3.

All commands of the management and data interfaces are detailed in Section 2.4. This set of commands is designed to support decentralized, centralized and hybrid scheduling solutions.

6top defines TSCH Information Elements (IEs) for neighbors nodes to negotiate scheduling cells in the TSCH schedule. The format of those is given in Section 2.5. Example data exchanges between neighbor nodes are illustrated in Section 2.6.

Section 2.7 defines how 6top gathers statistics (e.g., link quality, energy level, queue usage), and what commands an upper layer can use to configure and retrieve statistics.

6top can be configured to monitor the cells it has scheduled in order to detect cells with poor performance. It can automatically re-allocate those cells inside the TSCH schedule. This behavior is described in Section 2.8

2.2. Cell Model

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

b0 = Transmit

b1 = Receive

b2 = Shared

b3 = Timekeeping

b4-b7 = Reserved

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

6top allows an upper layer to schedule a cell at a specific slotOffset and channelOffset, in a specific slotframe. 6top follows the hard cell reservation process described in Section 2.6.5.

In addition, 6top allows an upper layer to schedule a certain amount of bandwidth to a neighbor, without having to specify the exact slotOffset(s) and channelOffset(s). 6top follows the soft cell reservation process described in Section 2.6.2. Once bandwidth is reserved, 6top is in charge of ensuring that this requirement is continuously satisfied, as described in Section 2.8. 6top dynamically reallocates cells if needed, and over-provisions if required.

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

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

Given this mechanism, 6top defines hard cells (which have been requested specifically) and soft cells (which can be reallocated

dynamically). The hard/soft flag is introduced by the 6top sublayer as an extension of LinkOption flags defined in [IEEE802154e]. This option is mandatory; all cells are either hard or soft.

With the addition of the Hard/Soft flag, the resulting flags are:

- b0 = Transmit
- b1 = Receive
- b2 = Shared
- b3 = Timekeeping
- b4 = Hard (1)/Soft (0)
- b5-b7 = Reserved

2.2.1. hard cells

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

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

slotOffset: the slotOffset for the cell.

channelOffset: the channelOffset for the cell.

LinkOption bitmap: bitmap as defined in Section 2.2, including the hard/soft bit which MUST be set to 1.

2.2.2. soft cells

A soft cell is a cell that can be reallocated by 6top dynamically. The hard/soft bit MUST be set to 0. This cell is installed by 6top given a specific bandwidth requirement. Soft cells are installed through the soft cell negotiation procedure described in Section 2.6.

2.3. Data Convey Model

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

TSCH is the fragment corresponding to the next active cell in the TSCH schedule.

6top Data Convey Model

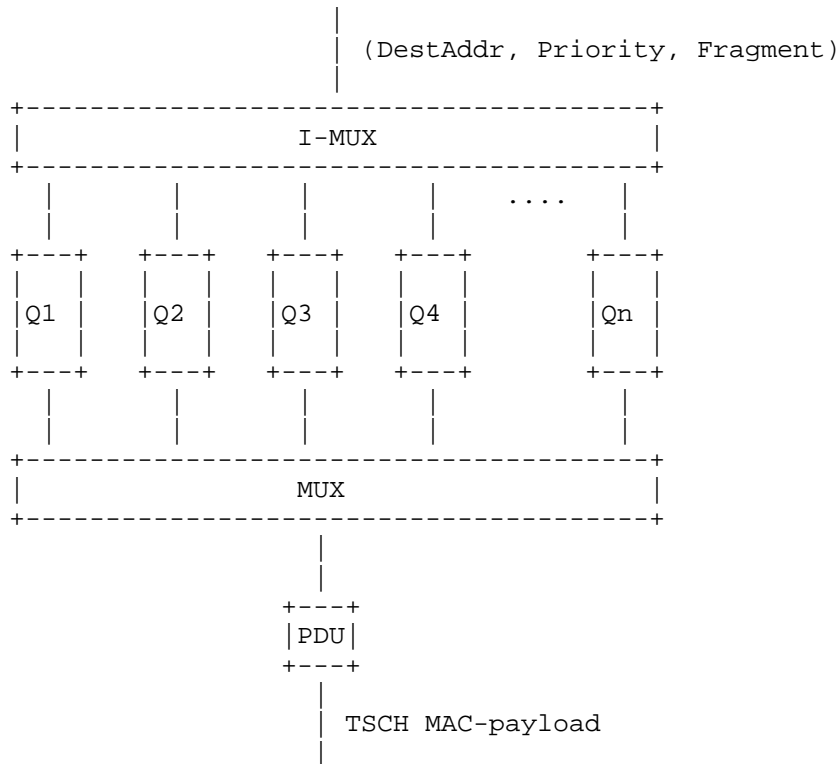


Figure 2

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

When 6top receives a packet to transmit through a `Send.data` command (Section 2.4.10), the I-MUX module selects a queue in which to insert it. If the packet's destination address is a unicast (resp. broadcast) address, it will be inserted into a unicast (resp. broadcast) queue.

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

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

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

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

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

Further rules can be configured to satisfy specific QoS requirements.

2.4. Commands

6top provides a set of commands as the interface with the higher layer. Most of these commands are related to the management of slotframes, cells and scheduling information. 6top also provides an interface allowing an upper layer to retrieve status information and statistics. This section describes the following commands provided by 6top.

CREATE.hardcell: Section 2.4.1.1

CREATE.softcell: Section 2.4.1.2

READ.cell: Section 2.4.1.3

UPDATE.cell: Section 2.4.1.4

DELETE.hardcell: Section 2.4.1.5

DELETE.softcell: Section 2.4.1.6

REALLOCATE.softcell: Section 2.4.1.7

CREATE.slotframe: Section 2.4.2.1

READ.slotframe: Section 2.4.2.2

UPDATE.slotframe: Section 2.4.2.3

DELETE.slotframe: Section 2.4.2.4

CONFIGURE.monitoring: Section 2.4.3.1

READ.monitoring: Section 2.4.3.2

CONFIGURE.statistics: Section 2.4.4.1

READ.statistics: Section 2.4.4.2

RESET.statistics: Section 2.4.4.3

CONFIGURE.eb: Section 2.4.5.1

READ.eb: Section 2.4.5.2

CONFIGURE.timesource: Section 2.4.6.1

READ.timesource: Section 2.4.6.2

CREATE.neighbor: Section 2.4.7.1

READ.all.neighbor: Section 2.4.7.2

READ.neighbor: Section 2.4.7.3

UPDATE.neighbor: Section 2.4.7.4

DELETE.neighbor: Section 2.4.7.5

CREATE.queue: Section 2.4.8.1

READ.queue: Section 2.4.8.2

READ.queue.stats: Section 2.4.8.3

UPDATE.queue: Section 2.4.8.4

DELETE.queue: Section 2.4.8.5

CONFIGURE.security: Section 2.4.9.1

CONFIGURE.security.macKeyTable: Section 2.4.9.2

CONFIGURE.security.macSecurityLevelTable:Section 2.4.9.3

Send.data: Section 2.4.10.1

Receive.data: Section 2.4.10.2

LabelSwitching.map: Section 2.4.11.1

LabelSwitching.unmap: Section 2.4.11.2

2.4.1. Cell Commands

6top provides the following commands to manage TSCH cells.

2.4.1.1. CREATE.hardcell

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

To create a hard cell, the upper layer specifies:

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

slotOffset: the slotOffset for the cell.

channelOffset: channelOffset for the cell.

LinkOption bitmap: bitmap as defined in Section 2.2

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

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

6top schedules the cell and marks it as a hard cell, indicating that it cannot reschedule this cell.

2.4.1.2. CREATE.softcell

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

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

number of cells: the required number of soft cells.

LinkOption bitmap: bitmap as defined in Section 2.2

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

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

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

6top is responsible for picking the exact slotOffset and channelOffset in the schedule, and ensure that the target node choose the same cell and TrackID. 6top marks these cells as soft cell, indicating that it will continuously monitor their performance and reschedule if needed.

6top deals with the allocation process by negotiation with the target node. The negotiation process is described in Section 2.6.2. The command returns the list of created cells defined by (slotframe ID, slotOffset, channelOffset). It fails if the required number of cells is higher than the available number of cells in the schedule. It fails if the negotiation with the target node fails. It fails if the LinkOption bitmap indicates that the cell(s) MUST be Hard.

2.4.1.3. READ.cell

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

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

slotOffset: the slotOffset for the cell.

channelOffset: the selected channelOffset for the cell.

LinkOption bitmap: bitmap as defined in Section 2.2

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

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

A read command can be issued for any cell, hard or soft.

2.4.1.4. UPDATE.cell

Update a hard cell, i.e., re-allocate it to a different slotOffset and/or channelOffset. Fails if the cell does not exist. Requires

both old (slotframe ID, slotOffset, channelOffset) and new (slotframe ID, slotOffset, channelOffset) as parameters. And, the type of cell, target node address and TrackID are the fields that cannot be updated. Soft cells MUST NOT be updated by the UPDATE.cell command. REALLOCATE.softcell (Section 2.4.1.7) MUST be used instead.

2.4.1.5. DELETE.hardcell

To remove a hard cell, the upper layer specifies:

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

slotOffset: the slotOffset for the cell.

channelOffset: the selected channelOffset for the cell.

This removes the hard cell from the node's schedule.

2.4.1.6. DELETE.softcell

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

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

number of cells: the number of cells to be removed

LinkOption bitmap: bitmap as defined in Section 2.2

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

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

In the case a soft cell wants to be re-allocated from the allocated cell so a hard cell can be installed instead, the REALLOCATE.softcell (Section 2.4.1.7) MUST be used.

2.4.1.7. REALLOCATE.softcell

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

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

slotOffset: the slotOffset for that cell.

channelOffset: the channelOffset for that cell.

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

2.4.2. Slotframe Commands

6top provides the following commands to manage TSCH slotframes.

2.4.2.1. CREATE.slotframe

Creates a new slotframe. Returns the slotframe ID that corresponds to its priority (SlotFrameHandle). The command requires:

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

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

2.4.2.2. READ.slotframe

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

slotframe ID: ID of the slotframe. (SlotFrameHandle)

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

Fails if the slotframe ID does not exist.

2.4.2.3. UPDATE.slotframe

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

slotframe ID: ID of the slotframe.

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

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

2.4.2.4. DELETE.slotframe

Deletes a slotframe. The command requires:

slotframe ID: ID of the slotframe.

Fails if the slotframe ID does not exist.

2.4.3. Monitoring Commands

Monitoring commands provide the means for upper layers to configure whether 6top must ensure the required bandwidth. This procedure is achieved through overprovisioning according to cell status feedback. Monitoring is also in charge of reallocating soft cells that are under the required QoS. The mechanism is described in Section 2.8.

2.4.3.1. CONFIGURE.monitoring

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

slotframe ID: ID of the slotframe.

target node address: the target neighbor address.

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

Fails if the slotframe ID does not exist.

2.4.3.2. READ.monitoring.status

Reads the current Monitoring status. Requires the following parameters.

slotframe ID: the ID of the slotframe.

target node address: the target neighbor address.

Returns the QoS levels for that Target node on that slotframe.

allocated_hard: Number of hard cells allocated.

allocated_soft: Number of soft cells allocated.

provisioned: the extra provisioned cells. 0 if CONFIGURE.qos enforce is DISABLE.

QoS: the current QoS. Including overprovisioned cells, i.e what bandwidth is being obtained including the overprovisioned cells.

RQoS: the real QoS without provisioned cells. What is the actual bandwidth without taking into account the overprovisioned cells.

Fails if the slotframe ID does not exist.

2.4.4. Statistics Commands

6top keeps track of TSCH statistics for upper layers to adapt correctly to medium changes. The exact metrics for statistics are out of scope but the present commands SHOULD be used to configure and read monitored information regardless of the specific metric.

2.4.4.1. CONFIGURE.statistics

Configures Statistics process. The command requires:

slotframe ID: ID of the slotframe. If empty monitors all slotframe IDs

slotOffset: specific slotOffset to be monitored. If empty all timeslots are monitored

channelOffset: specific channelOffset to be monitored. If empty all channels are monitored.

target node address: the target neighbor address. If empty, all neighbor nodes are monitored.

metric: metric to be monitored. This MAY be PDR, ETX, queuing statistics, energy-related metrics, etc.)

window: time window to be considered for the calculations. If 0 all historical data is considered.

enable: Enables statistics or disables them.

Fails if the slotframe ID does not exist. The statistics service can be configured to retrieve statistics at different levels. For example to aggregate information by slotframe ID, or to retrieve statistics for a particular timeslot, etc. The CONFIGURE.statistics enables flexible configuration and supports empty parameters that will force 6top to conduct statistics on all members of that dimension. For example, if ChannelOffset is empty and metric is set as PDR, then, 6top will conduct the statistics of PDR on all of channels.

2.4.4.2. READ.statistics

Reads a metric for the specified dimension. Information is aggregated according to the parameters. The command requires:

slotframe ID: ID of the slotframe. If empty aggregates information of all slotframe IDs

slotOffset: the specific slotOffset for which the information is required. If empty all timeslots are aggregated

channelOffset: the specific channelOffset for which the information is required. If empty all channels are aggregated.

target node address: the target neighbor address. If empty all neighbor addresses are aggregated.

metric: metric to be read.

Returns the value for the requested metric.

Fails if empty metric or metric does not exists.

2.4.4.3. RESET.statistics

Resets the gathered statistics. The command requires:

slotframe ID: ID of the slotframe. If empty resets the information of all slotframe IDs

slotOffset: the specific slotOffset for which the information wants to be reset. If empty statistics from all timeslots are reset

channelOffset: the specific channelOffset for which the information wants to be reset. If empty all statistics for all channels are reset.

target node address: the target neighbor address. If empty all neighbor addresses are aggregated.

metric: metric to be reset.

Fails if empty metric or metric does not exists.

2.4.5. Network Formation Commands

EBs need to be configured, including their transmission period, the slotOffset and channelOffset that they SHOULD be sent on, and the join priority they contain. The parameters for that command are optional and enable flexible configuration of EBs. If slotframe ID is specified, the EBs will be configured to use that specific slotframe; if not, they will use the first slotframe where the configured slotOffset is allocated. The slotOffset enforces the EB to a specific timeslot. In case slotOffset parameter is not present, the EB is sent in the first available transmit timeslot. In case channelOffset parameter is not set, the EB is configured to use the first available channel.

2.4.5.1. CONFIGURE.eb

Configures EBs. The command requires:

slotframe ID: ID of the slotframe where the EBs MUST be sent. Zero if any slotframe can be used.

slotOffset: the slotOffset where the EBs MUST be sent. Zero if any timeslot can be used.

channelOffset: the channelOffset where the EBs MUST be sent. Zero if any channelOffset can be used.

period: the EBs period, in seconds.

Expiration: when the EBs periodicity will stop. If Zero the period never stops.

priority: the joining priority model that will be used for advertisement. Joining priority MAY be for example SAME_AS_PARENT, RANDOM, BEST_PARENT+1 or DAGRANK(rank) as described in in [I-D.vilajosana-6tisch-minimal].

Fails if the tuple (slotframe ID, slotOffset, channelOffset) is already scheduled.

2.4.5.2. READ.eb

Reads the EBs configuration. No parameters are required.

Returns the current EBs configuration for that slotframe, which contains:

slotframe ID: the slotframe where the EB is being sent.

slotOffset: the slotOffset where the EBs is being sent.

channelOffset: the channelOffset the EBS is being sent on.

period: the EBS period.

Expiration: when the EBS periodicity stops. If 0 the period never stops.

priority: the joining priority that this node advertises.

Fails if the slotframe ID does not exist.

2.4.6. Time Source Neighbor Commands

Commands to select time source neighbors.

2.4.6.1. CONFIGURE.timesource

Configures the Time Source Neighbor selection process. More than one time source neighbor can be selected. The command requires:

selection policy: The policy used to select the time source neighbor. The policy MAY be for example ALL_PARENTS, BEST_CONNECTED, LOWEST_JOIN_PRIORITY, etc.

Fails if any of the time source neighbors do not exist or it is not reachable.

2.4.6.2. READ.timesource

Retrieves information about the time source neighbors of that node. The command does not require any parameter.

Returns the following information for each of the time sources:

target node: address of the time source neighbor.

statistics: includes for example minimum, maximum, average time correction for that time source neighbor

policy: the used policy

Fails if the slotframe ID or no time source neighbors exist.

2.4.7. Neighbor Commands

Commands to manage neighbor table. The commands SHOULD be used by the upper layer to query the neighbor related information and by the lower layer to keep track of neighbors information.

2.4.7.1. CREATE.neighbor

Creates an entry for a neighbor in the neighbor table.

neighbor address: The address of the neighbor.

neighbor stats: for example, RSSI of the last received packet from that neighbor, ASN when that neighbor has been added, etc.

Returns whether the neighbor is created or not.

2.4.7.2. READ.all.neighbor

Returns the list of neighbors of that node. Fails if empty. For each neighbor in the list it returns:

neighbor address: The address of the neighbor.

neighbor stats: for example, RSSI of the last received packet from that neighbor, ASN when that neighbor has been added, packets received from that neighbor, packets sent to it, etc.

2.4.7.3. READ.neighbor

Returns the information of a specific neighbor of that node specified by its neighbor address. Fails if it does not exist. For that neighbor it returns:

neighbor address: The address of the neighbor.

neighbor stats: for example, RSSI of the last received packet from that neighbor, ASN when that neighbor has been added, packets received from that neighbor, packets sent to it, etc.

2.4.7.4. UPDATE.neighbor

Updates an entry for a neighbor in the neighbor table. Fails if the neighbor does not exist. Updates stats parameters. Requires:

neighbor address: The address of the neighbor.

neighbor stats: for example, RSSI of the last received packet from that neighbor, ASN when that neighbor has been added, etc.

Returns whether the neighbor is updated or not.

2.4.7.5. DELETE.neighbor

Deletes a neighbor given its address. Fails if the neighbor does not exist.

2.4.8. Queueing Commands

Queues need to be configured. This includes queue length, retransmission policy, discarding of packets, etc.

2.4.8.1. CREATE.queue

Creates and Configures Queues. The command SHOULD be applied for each required queue. The command requires:

txqlength: the desired transmission queue length.

rxqlength: the desired reception queue length.

numrtx: number of allowed retransmissions.

age: discard packet according to its age on the queue. 0 if no discards are allowed.

rtxbackoff: retransmission backoff in number of slotframes. 0 if next available timeslot wants to be used.

statswindow: window of time used to compute stats.

queue priority: the priority of this queue.

TrackIDs: a set of TrackIDs. While it is empty, no specific track is associated with the queue

Returns the queue ID.

2.4.8.2. READ.queue

Reads the queue configuration. Requires the queue ID.

The command returns:

txqlength: the transmission queue length.

rxqlength: the reception queue length.

numrtx: number of allowed retransmissions.

age: maximum age of a packet before being discarded. 0 if no discards are allowed.

rtxbackoff: retransmission backoff in number of slotframes. 0 if next available timeslot is used.

2.4.8.3. READ.queue.stats

Reads the queue stats. Requires queue ID.

The command returns:

txqlengthstats: average, maximum, minimum length of the transmission queue.

rxqlengthstats: average, maximum, minimum length of the reception queue.

numrtxstats: average, maximum, minimum number of retransmissions.

agestats: average, maximum, minimum age of a packet in the queue.

rtxbackoffstats: average, maximum, minimum retransmission backoff.

queue priority: the priority of this queue.

TrackIDs: a set of TrackIDs.

2.4.8.4. UPDATE.queue

Update a Queue. The command requires:

queueid: the queue ID.

txqlength: the desired transmission queue length.

rxqlength: the desired reception queue length.

numrtx: number of allowed retransmissions.

age: discard packet according to its age on the queue. 0 if no discards are allowed.

rtxbackoff: retransmission backoff in number of slotframes. 0 if next available timeslot wants to be used.

statswindow: window of time used to compute stats.

queue priority: the desired priority of this queue.

TrackIDs: the desired set of TrackIDs.

2.4.8.5. DELETE.queue

Deletes a Queue. The command requires the queue ID. All packets in the queue are discarded and the queue is deleted.

2.4.9. Security Commands

The following commands are used to manage underlying layer security. In that case 6top acts as delegating interface to the security attributes defined in the MAC PIB ([IEEE802154]).

2.4.9.1. CONFIGURE.security

Enables/Disables Security and configures the MAC PIB. The command requires:

enable: enables underlying layer security.

macAutoRequestSecurityLevel: the security level used for automatic data requests as described by table 60 in [IEEE802154].

macAutoRequestKeyIdMode: the key identifier mode used for automatic data requests as described by table 60 in [IEEE802154].

macAutoRequestKeySource: the originator of the key for automatic data requests as described by table 60 in [IEEE802154].

macAutoRequestKeyIndex: the index of the key used for automatic data requests as described by table 60 in [IEEE802154].

macDefaultKeySource: the originator of the default key used for key identifier mode 0x01 as described by table 60 in [IEEE802154].

macPANCordinatorExtendedAddress: Address of the PAN coordinator as described by table 60 in [IEEE802154].

macPANCordinatorShortAddress: Short address of the PAN coordinator as described by table 60 in [IEEE802154].

2.4.9.2. CONFIGURE.security.macKeyTable

Configures Security Keys. The command requires:

KeyIdLookupList: list of keyIdLookupDescriptor Entries as defined by table 61 in [IEEE802154].

DeviceDescriptorHandleList: Implementation specific list of devices that are using this key. As defined by table 61 in [IEEE802154].

KeyUsageList: List of slotframe types on which this key is being used as specified by table 61 in [IEEE802154].

Key: 16 octets key. As specified by table 61 in [IEEE802154].

2.4.9.3. CONFIGURE.security.macSecurityLevelTable

Configures the set of security levels. The command requires:

FrameType: Slotframe type as defined by table 63 in [IEEE802154].

Command Identifier: The command identifier as defined by table 63 in [IEEE802154].

Security Minimum: The minimum required security level as specified by table 63 in [IEEE802154].

Device Override Security Minimum: whether the minimum security level can be overridden as specified by table 63 in [IEEE802154].

Allowed Security Levels: the key identifier field that identifies the key that is being used as specified by table 63 in [IEEE802154].

2.4.9.4. Security Command Behavior

6top offers the interface to upper layers so underlying MAC layer can be configured. In that sense, 6top only delegates the functionalities to the MAC security services. For more details Section 7 on [IEEE802154] and its amendments on [IEEE802154e] SHOULD be referred.

2.4.10. Data Commands

2.4.10.1. Send.data

The command used by upper layers to queue a packet so underlying TSCH sends it. According to the specific priority, the packet is pushed into a Queue with the equivalent priority or following a criteria out of scope. Once a packet is inserted into a queue it waits to be transmitted by TSCH according to the model defined in Section 2.3. If the queue is full or destination address is not a L2 neighbor of the node, failure to enqueue will be indicated to the caller.

The required parameters are:

src address: L2 address

dest address: L2 unicast or broadcast address

priority: packet priority, usually is consistent with queue priority

message length: the length of the message

message: control message or data message

securityLevel:As defined by [IEEE802154e].

2.4.10.2. Receive.data

The command is invoked whenever a packet is received and inserted into a reception queue. The method acts as a callback function to notify to the upper layers the received message. Upper layers MUST terminate this indication.

The function has the following parameters:

src address: L2 source address

dest address: L2 unicast or broadcast destination address

priority: packet priority, usually is consistent with queue priority

message length: the length of the message.

message: control message or data message

2.4.11. Label Switching Commands

2.4.11.1. LabelSwitching.map

The command used by an upper layer to map an input cell or a bundle of input cells to an output cell or a bundle of output cells. 6top stores this mapping and makes sure that the packets are forwarded at the specific output cell/bundle. Label Switching is enabled by the specified bundle as soon as the mapping is installed.

The required parameters are:

input cells: list of input cells (one or more cells in a bundle). Each input cells is described by an unique tuple (slotOffset, channelOffset, destination address).

output cells: list of output cells (one or more cells in a bundle). Each output cells is described by an unique tuple (slotOffset, channelOffset, destination address).

load balancing policy: A policy for load balance cell usage. The policy is out of scope, however an example can be use ROUND ROBIN policy within the cells of the same bundle.

2.4.11.2. LabelSwitching.unmap

The command used by upper layers to unmap one input cell or a bundle of input cells to an output cell or a bundle of output cells. The mapping is removed from the state kept by 6top.

The required parameters are:

input cells: list of input cells (one or more cells in a bundle). Each input cells is described by an unique tuple (slotOffset, channelOffset, destination address).

output cells: list of output cells (one or more cells in a bundle). Each output cells is described by an unique tuple (slotOffset, channelOffset, destination address).

2.5. Message Formats

6top has to negotiate the scheduling of soft cells with neighbor nodes. This negotiation happens through 6top-specific TSCH Information Elements, the format of which is defined in this section. For completeness, this section also details the formats of the IEs already defined in [IEEE802154e] and presented here without modification.

6top messages can contain one or more IEs. Section 2.5.1 defines the different IEs used by 6top, both the ones used without modification from [IEEE802154e], and the new ones defined by this document. Section 2.5.2 shows how several IEs are assembled to form the different frames used by 6top.

2.5.1. Information Elements

[IEEE802154e] defines Information elements (IEs). IEs are formatted data objects consisting of an ID, a length, and a data payload used to pass data between layers or devices. [IEEE802154e] defines Header IEs and Payload IEs; 6top only uses Payload IEs. A Payload IE includes one or more IEs, and ends with a termination IE (ID = 0xf, see [IEEE802154e]).

6top uses the following Information Elements, some defined in [IEEE802154e], others introduced in this document.

Defined in [IEEE802154e] and used by 6top without modification:

TSCH Synchronization IE (Section 2.5.1.1)

TSCH Slotframe and Link IE (Section 2.5.1.2)

TSCH Timeslot Template IE (Section 2.5.1.3)

TSCH Channel Hopping IE (Section 2.5.1.4)

Defined by 6top:

6top Opcode IE (Section 2.5.1.5)

6top Bandwidth IE (Section 2.5.1.6)

6top TrackID IE (Section 2.5.1.7)

6top Generic Schedule IE (Section 2.5.1.8)

2.5.1.1. TSCH Synchronization IE

A Synchronization IE (SyncIE) contains Information allowing a node to synchronize to a TSCH network, including the current ASN and a join priority. Synchronization IE MUST be included in all TSCH Enhanced Beacons.

6top re-uses this IE as defined in [IEEE802154e].

Format of a TSCH Synchronization IE (SyncIE).

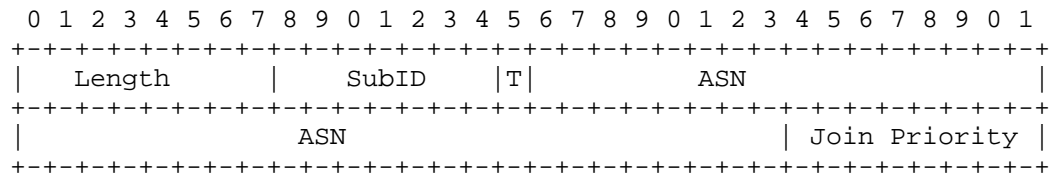


Figure 3

Length=6

SubID=0x1a

T=0, i.e., short type

ASN (5 octets) contains the Absolute Slot Number corresponding to the timeslot in which the TSCH Enhanced Beacon is sent.

The Join Priority can be used by a joining device to select among beaconing devices when multiple beacons are heard. The PAN coordinator's join priority is zero. A lower value of join priority indicates that the device is the preferred one to connect to. As suggested by [I-D.vilajosana-6tisch-minimal], the beaconing device's join priority is its DAGRank(rank).

2.5.1.2. TSCH Slotframe and Link IE

The Slotframe and Link IE (FrameAndLinkIE) contains one or more slotframes and their respective cells that a beaconing device advertises to allow other devices to join the network.

6top re-uses this IE as defined in [IEEE802154e].

Format of a TSCH Slotframe and Link IE (FrameAndLinkIE).

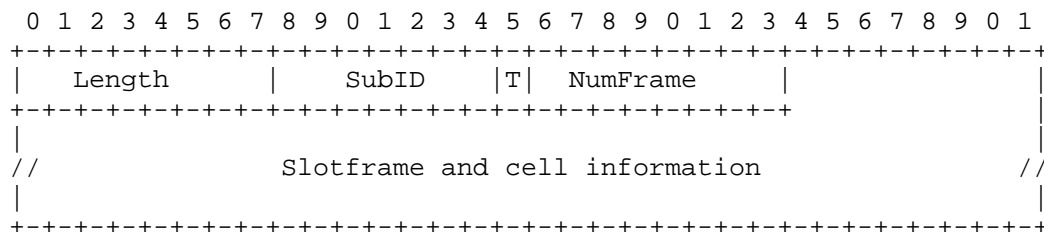


Figure 4

Length=variable

SubID=0x1b

T=0, i.e., short type

NumFrame is set to the total number of slotframe descriptors contained in the TSCH Enhanced Beacon.

Format of a slotframe descriptor.

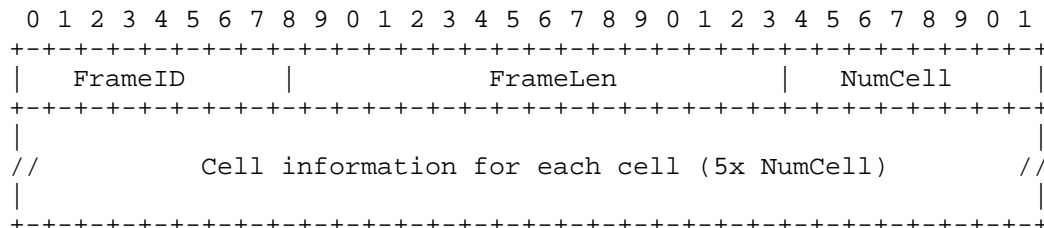


Figure 5

The FrameID field shall be set to the slotframeHandle that uniquely identifies the slotframe.

The FrameLen field shall be set to the size of the slotframe in number of timeslots.

The NumCell field shall be set to the number of cells that belong to the specific slotframe identified by the slotframeHandle.

Format of a Cell information.

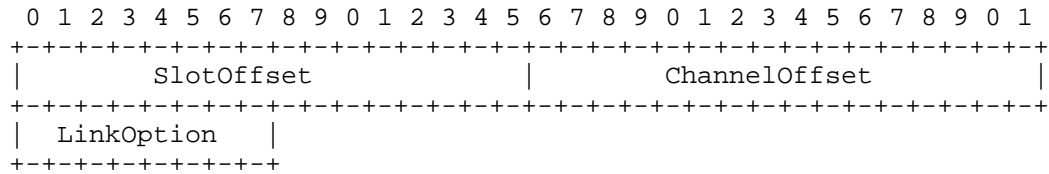


Figure 6

SlotOffset shall be set to the slotOffset of this cell.

ChannelOffset shall be set to the channelOffset of this cell.

LinkOption indicates whether this cell is a TX cell, an RX cell, or a SHARED TX cell, whether the device to which it is being linked is to be used for clock synchronization, and whether this cell is hard cell.

2.5.1.3. TSCH Timeslot Template IE

Timeslot Template IE (SlotTemplateIE) defines Timeslot template being used by the TSCH device.

6top re-uses this IE as defined in [IEEE802154e].

Format of a TSCH Timeslot Template IE (SlotTemplateIE).

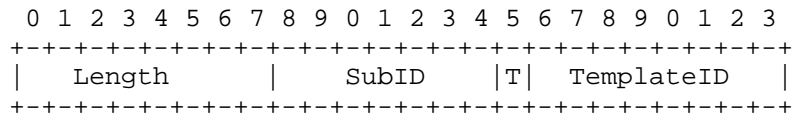


Figure 7

Length=1

SubID=0x1c

T=0, i.e., short type

TemplateID shall be set to a Timeslot template handle. The full timeslot template, which contains the macTimeslotTemplate of TSCH (total 25 octets), MAY be included.(see [IEEE802154e]).

2.5.1.4. TSCH Channel Hopping IE

Channel Hopping IE (ChHoppingIE) defines the Hopping Sequence being used by the TSCH device.

6top re-uses this IE as defined in [IEEE802154e].

Format of a TSCH Channel Hopping IE (ChHoppingIE).

```

 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+---+---+---+---+---+---+---+---+---+---+---+---+
|      Length      | SubID |T| HopSequenceID |
+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 8

Length=1

SubID=0x09

T=1, i.e., long type

HopSequenceID shall be set to a Hopping Sequence handle. The full Hopping Sequence information MAY be included. (see [IEEE802154e]).

2.5.1.5. 6top Opcode IE

6top Opcode IE (OpcodeIE) defines operation codes of packets in 6top sublayer.

This IE is not present in [IEEE802154e] and is defined by 6top.

Format of a 6top Opcode IE (OpcodeIE).

```

 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+---+---+---+---+---+---+---+---+---+---+---+---+
|   Length   |   SubID   |T|   OpcodeID   |
+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 9

Length=1

SubID=0x41

T=0, i.e., short type

OpcodeID field shall be set to one of the following codes.

0x00: Reserve Soft Cell Request

0x01: Reserve Soft Cell Response

0x02: Remove Soft Cell Request

0x03: Reserve Hard Cell Request

0x04: Remove Hard Cell Request

2.5.1.6. 6top Bandwidth IE

Bandwidth IE (BwIE) defines the number of cells to be reserved or actually be reserved.

This IE is not present in [IEEE802154e] and is defined by 6top.

Format of a 6top Bandwidth IE (BwIE).

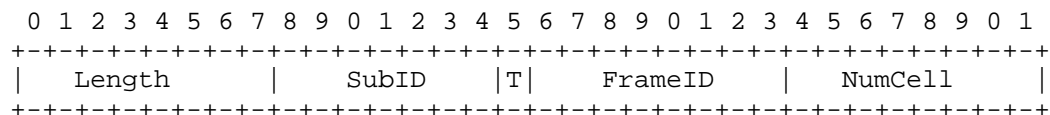


Figure 10

Length=2

SubID=0x42

T=0, i.e., short type

FrameID MAY be set to the SlotFrameHandle to identify the slotframe from which cells are reserved. FrameID field MAY be set to NOP, which means no specific slotframe is associated.

NumCell shall be set to the number of cells. When BwIE is combined with the OpcodeID of Reserve Soft Cell Request, NumCell presents how many cells are required to reserve; and when BwIE is combined with the OpcodeID of Reserve Soft Cell Response, NumCell presents how many cells are reserved successfully.

2.5.1.7. 6top TrackID IE

TrackID IE (TrackIdIE) describes the track which the reserved/removed cell(s) are associated with.

This IE is not present in [IEEE802154e] and is defined by 6top.

Format of a 6top TrackID IE (TrackIdIE).

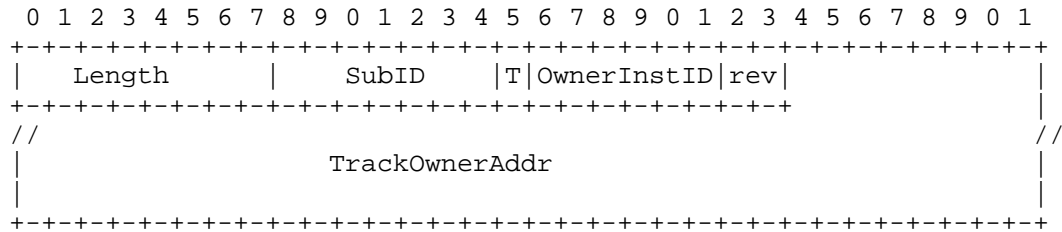


Figure 11

Length=3 or 7. When length=3, TrackOwnerAddr is 2 bytes short address, and when length=7, TrackOwnerAddr is 6 bytes long address.

SubID=0x43

T=0, i.e., short type

The combination of TrackOwnerAddr and OwnerInstId represents a specific TrackID.

2.5.1.8. 6top Generic Schedule IE

Generic Schedule IE (ScheduleIE) describes cell sets. In different packets, ScheduleIE represents different information. See Section 2.5.2 for more detail.

This IE is not present in [IEEE802154e] and is defined by 6top.

Format of a 6top Generic Schedule IE (ScheduleIE).

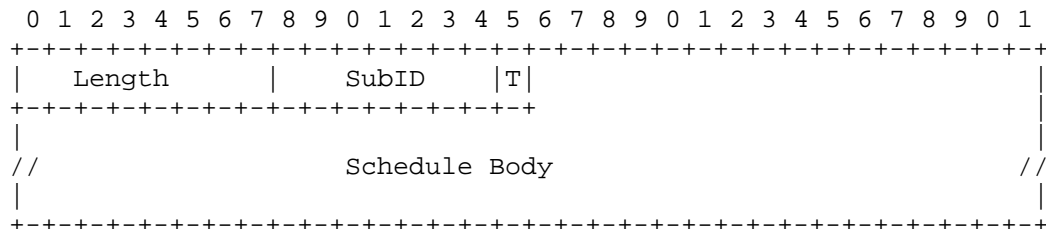


Figure 12

Length=variable

SubID=0x44

T=0, i.e., short type

Schedule Body carries one or more schedule object. An object MAY carry a TLV (Type-Length-Value), which MAY itself comprise other TLVs. TLV format is as follows. Type: 1 byte, Length: 1 byte, Value: variable

The following are some examples of schedule object TLV.

Example 1. Cell Set TLV

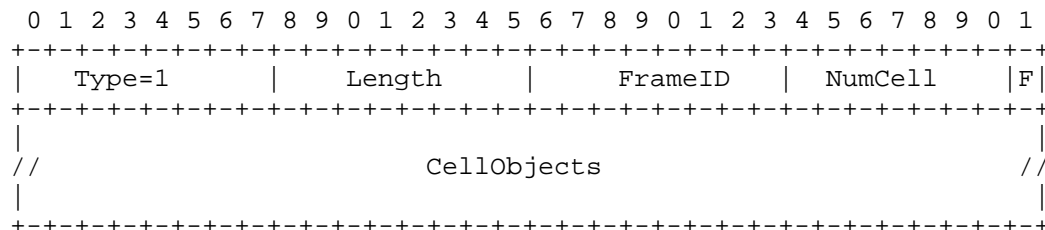


Figure 13

FrameID shall be set to the slotframeHandle that uniquely identifies the slotframe.

NumCell shall be set to the number of cells that belong to the specific slotframe identified by the slotframeHandle.

F=1 means the specified cells equals to what are listed in CellObjects, and F=0 means the specified cells equals to what are not listed in CellObjects.

CellObjects carries the information for one or more cells, including SlotOffset, ChannelOffset, LinkOption (Figure 6).

Example 2. Schedule Matrix TLV

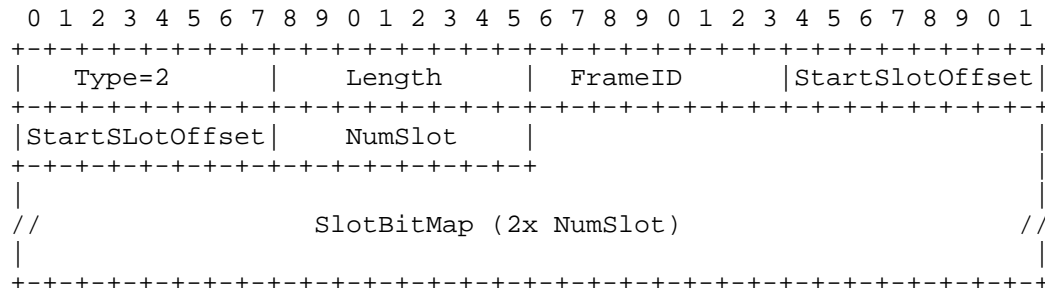


Figure 14

FrameID field MUST be set to the slotframeHandle that uniquely identifies the slotframe.

StartSlotOffset field (2 octets) MUST be set to the slotOffset in the specific slotframe identified by the slotframeHandle.

NumSlot field MUST be set to the number of timeslots from StartSlotOffset in the specific slotframe identified by the slotframeHandle.

SlotBitMap (per timeslot) indicates for the given timeslot which channels are specified. For the 16 channels in 2.4GHz band, 2-octets are used to indicate which channel is specified. For example, given a timeslot and a SlotBitmap with value (10001000,00010000); the bitmap represents that ChannelOffset-0, ChannelOffset-4, ChannelOffset-11 are specified.

2.5.2. Packet Formats

This section describes the packets used in 6top to form a network, reserve/maintain bandwidth using soft cells, and reserve/remove hard cells in both the transmitter side and receiver sides. Each of these packets uses one or more IEs defined in Section 2.5.1.

2.5.2.1. TSCH Enhanced Beacon

The TSCH Enhanced Beacon is used to announce the presence of the network and allows new nodes to join. It is an Enhanced Beacon packet defined in [IEEE802154e] with the following Payload IEs:

TSCH Synchronization IE (Section 2.5.1.1)

TSCH Timeslot Template IE (Section 2.5.1.3)

TSCH Channel Hopping IE (Section 2.5.1.4)

TSCH Slotframe and Link IE (Section 2.5.1.2)

Payload IE of TSCH Enhanced Beacon Packet

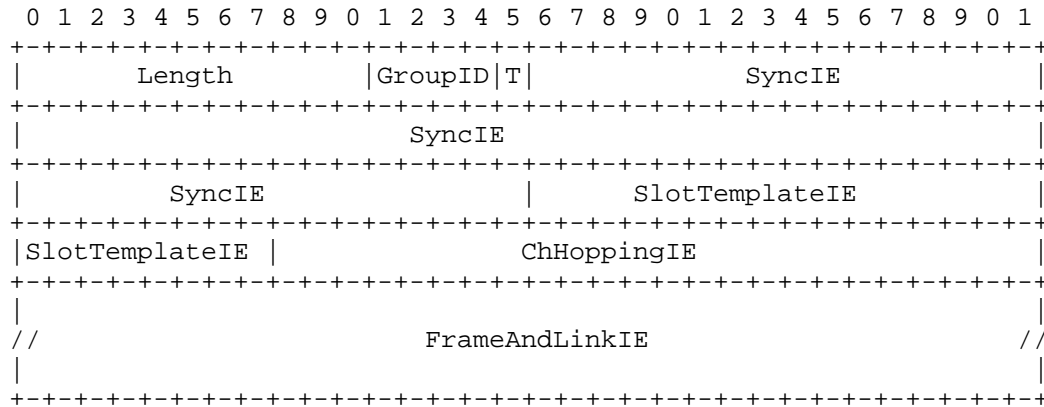


Figure 15

Length=variable

GroupID=0x1, i.e., MLME IE

T=1, i.e., payload IE

See Section 2.5.1.1, Section 2.5.1.3, Section 2.5.1.4, Section 2.5.1.2 for SyncIE, SlotTemplateIE, ChHoppingIE and FrameAndLinkIE.

2.5.2.2. Soft Cell Reservation Request

A Soft Cell Reservation Request packet is a DATA packet defined in [IEEE802154e] with the following payload IE.

Payload IE of Soft Cell Reservation Request

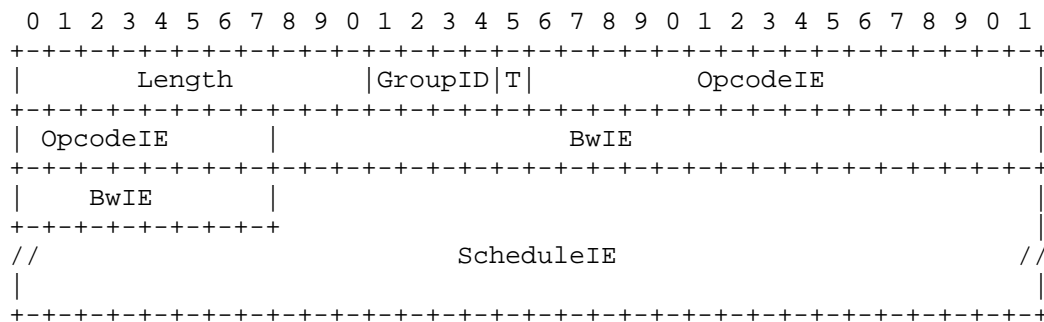


Figure 16

Length=variable

GroupID=0x1, i.e., MLME IE

T=1, i.e., payload IE

The OpcodeID field in the 3-octet OpcodeIE SHOULD be set to 0x00, indicates Reserve Soft Cell Request operation.

The NumCell field in 4-octet BwIE SHOULD be set to the number of cells needed to be reserved.

The ScheduleIE specifies a candidate cell set, from which the cells SHOULD be reserved. ScheduleIE MAY be empty, means there is no constrain on which cells SHOULD not be reserved.

In addition, TrackIdIE can be added in the packet to associate the reserved soft cells to a specific TrackID.

2.5.2.3. Soft Cell Reservation Response

Soft Cell Reservation Response is a DATA packet defined in [IEEE802154e] with the following payload IE.

Payload IE of Soft Cell Reservation Response

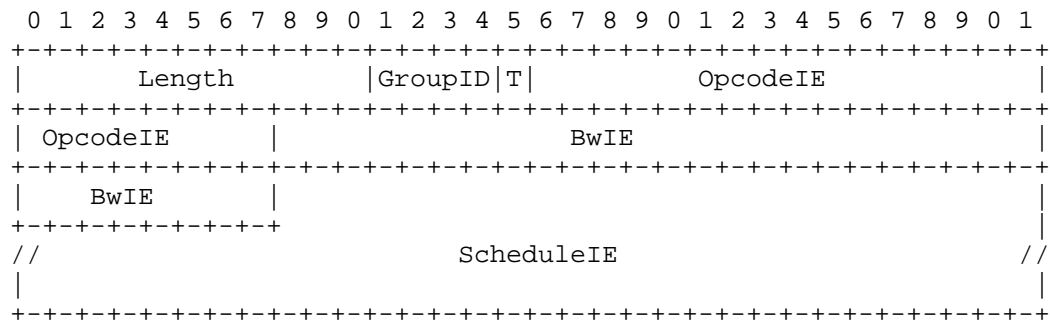


Figure 17

Length=variable

GroupID=0x1, i.e., MLME IE

T=1, i.e., payload IE

The OpcodeID field in the 3-octet OpcodeIE SHOULD be set to 0x01, indicates Reserve Soft Cell Response operation.

The NumCell field in 4-octet BwIE SHOULD be set to the number of cells which have been reserved successfully.

The ScheduleIE SHOULD specify all of the cells which have been reserved successfully.

In addition, TrackIdIE can be added in the packet to associate the reserved soft cells to a specific TrackID.

2.5.2.4. Soft Cell Remove Request

Soft Cell Remove Request is a DATA packet defined in [IEEE802154e] with the following payload IE.

Payload IE of Soft Cell Remove Request

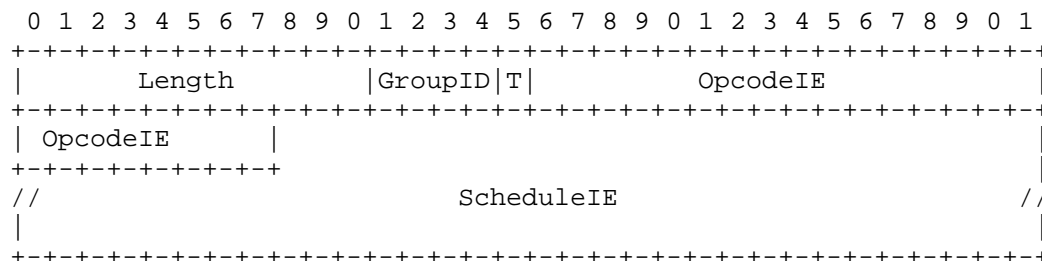


Figure 18

Length=variable

GroupID=0x1, i.e., MLME IE

T=1, i.e., payload IE

The OpcodeID field in the 3-octet OpcodeIE SHOULD be set to 0x02, indicates Remove Soft Cell Request operation.

The ScheduleIE SHOULD specify all the cells that need to be removed.

2.5.2.5. Hard Cell Reservation Request

Hard Cell Reservation Request packet is a DATA packet defined in [IEEE802154e] with the following payload IE.

Payload IE of Hard Cell Reservation Request

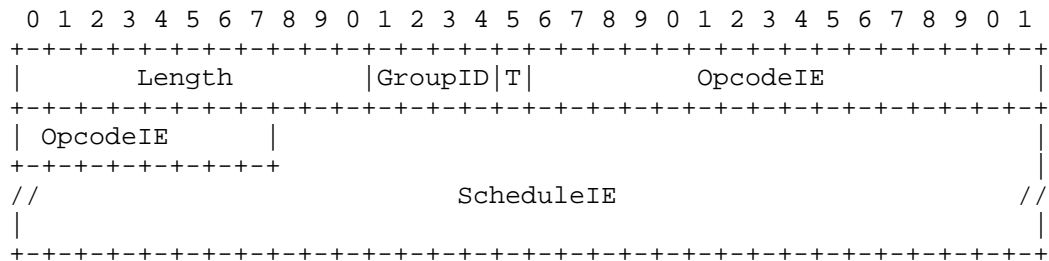


Figure 19

Length=variable

GroupID=0x1, i.e., MLME IE

T=1, i.e., payload IE

The OpcodeID field in the 3-octet OpcodeIE SHOULD be set to 0x03, indicates Reserve Hard Cell Request operation.

The ScheduleIE SHOULD specify all the cell that need to be reserved.

In addition, TrackIdIE can be added in the packet to associate the reserved hard cells to a specific TrackID.

2.5.2.6. Hard Cell Remove Request

Hard Cell Remove Request is a DATA packet defined in [IEEE802154e] with the following payload IE.

Payload IE of Hard Cell Remove Request

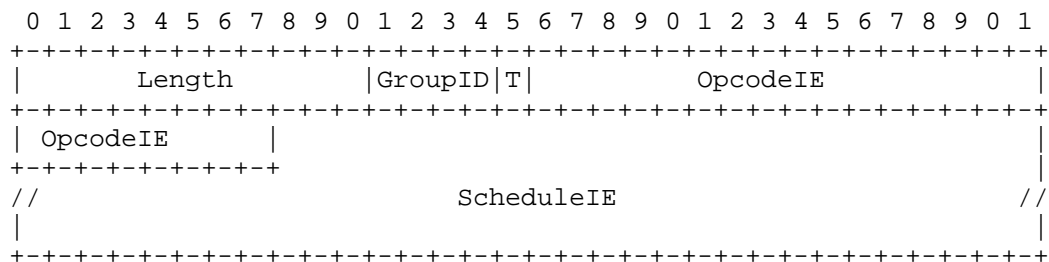


Figure 20

Length=variable

GroupID=0x1, i.e., MLME IE

T=1, i.e., payload IE

The OpcodeID field in the 3-octet OpcodeIE SHOULD be set to 0x04, indicates Remove Hard Cell Request operation.

The ScheduleIE SHOULD specify all the cells that need to be removed.

2.6. Time Sequence

6top neighbors exchange 6top-specific packets in the following cases, each detailed in a subsection.

- Network formation (Section 2.6.1)

- Creating soft cells (Section 2.6.2)

- Deleting soft cells (Section 2.6.3)

- Maintaining soft cells (Section 2.6.4)

- Creating hard cells (Section 2.6.5)

- Deleting hard cells (Section 2.6.6)

2.6.1. Network Formation

Network formation consists of two processes: joining and maintenance.

2.6.1.1. Joining

A node already in the network sends out TSCH Enhanced Beacons periodically.

When a node is joining an existing network, it listens for TSCH Enhanced Beacons. After collecting one or more TSCH Enhanced BEACONS (the format of which is detailed in Section 2.5.2.1), the joining node MUST do the following.

- Initialize a neighbor table. Establish a neighbor table and record all of the information described in the TSCH Enhanced BEACONS as its initial schedule with those neighbors.

- Select a time source neighbor. According to the Joining Priority described by SyncIEs, the joining node chooses time source neighbors. 6top does not specify the criteria to choose time source neighbors from the Enhanced BEACONS.

Select cells for Enhanced Beacons. The joining node selects one or more cells to indicate in its own Enhanced Beacons, which MAY be the same as the cells used by its neighbors for Enhanced Beacon broadcast, and record those cell(s) into the TSCH schedule with LinkType=ADVERTISING.

Its Enhanced Beacons SHOULD include the cell(s) selected for EB purposes. The EB cells MUST be configured with LinkOption to "Receive" and "Timekeeping", telling its neighbors that the cell is used for broadcast.

Start broadcasting Enhanced Beacon and communicate with neighbors.

2.6.1.2. Maintenance

Nodes MAY broadcast Enhance Beacons on the cells marked with LinkType=ADVERTISING, and listen for Enhanced Beacons from neighbors on the cells with LinkOption = "Receive" and "Timekeeping". If a cell with LinkType=ADVERTISING has both the "Receive" and "Timekeeping" LinkOptions set, which means that the cell is shared by neighbors and itself for broadcasting, then broadcasting Enhanced Beacon has higher priority.

Whenever a node receives an Enhanced Beacon, it SHOULD update its schedule if there is a difference regarding to the cells used for synchronizing with the advertiser of the Enhanced Beacon.

2.6.2. Creating soft cells

The upper layer instructs 6top to schedule one or more soft cells by calling the Create soft cell command. This command can also be called by the monitoring process internal to 6top.

When receiving a Create soft cell command, Node A's 6top sublayer forms a Soft Cell Reservation Request packet which includes the BwIE and ScheduleIE Information Elements. The BwIE indicates the number of cells to be reserved (N1); the ScheduleIE indicates set of a candidate cells from which the new cells SHOULD be selected. If the ScheduleIE is empty, Node A indicates there is no constraint on cell selection.

The Soft Cell Reservation Request is sent to the neighbor (Node B) with whom new cells need to be scheduled. After receiving the Soft Cell Reservation Request, Node B selects the cells from the candidate cell set defined by the ScheduleIE in the Soft Cell Reservation Request, and forms a Soft Cell Reservation Response packet. In the Cell Reservation Response packet, the BwIE indicates the number of

cells actually being reserved (N2); the ScheduleIE indicates those reserved cells. If N2 is smaller than N1, node B indicates to node A that there are not enough qualified cells to be reserved. Node B MUST record the reserved cells into its local schedule when sending the Soft Cell Reservation Response. After receiving the Soft Cell Reservation Response, Node A MUST record the reserved cells into its local schedule.

The policy to build a candidate cell set and the policy to select cells from the candidate cell set to reserve are out of scope.

The format of Schedule Body is flexible. For example, Node A can use Cell Set TLV defined in Figure 13 with field 'F' set to '0', and the CellObjects includes all of the cells being used by Node A. In another word, the cell candidate set is all of the cells not being included in the list defined by CellObjects.

The behavior of the nodes when the soft cells negotiation fails is out of scope.

2.6.3. Deleting soft cells

The upper layer instructs 6top to delete one or more soft cells by calling the Delete soft cell command (Section 2.4.1.6). This command can also be called by the monitoring process internal to 6top (Section 2.8).

When receiving a Delete soft cell command, Node A's 6top sublayer selects cells to be removed from its local schedule, and creates a Soft Cell Remove Request, which includes a ScheduleIE Information Element. The ScheduleIE indicates which specific cells to remove with a neighbor (Node B). The cells specified in the ScheduleIE SHOULD be removed from local schedule of Node A when the Soft Cell Remove Request is sent to Node B. When receiving the Soft Cell Remove Request, the cells specified in the ScheduleIE SHOULD be removed from the local schedule of Node B.

The policy to select cells corresponding to a Delete soft cell command is out of scope.

2.6.4. Maintaining soft cells

The monitoring process internal to 6top (Section 2.8) is responsible for monitoring and re-scheduling soft cells to meet some QoS requirements. The monitoring process MAY issue a soft cell Maintenance command, which indicate a set of cells to be re-allocated in the TSCH schedule.

When receiving a soft cell Maintenance command, 6top initializes a Soft Cell Remove Request (Section 2.6.3) with the neighbor in question, followed by a Soft Cell Reservation Request (Section 2.6.2).

2.6.5. Creating hard cells

The upper layer instructs 6top to create one or more hard cells by calling the Create hard cell command.

When receiving a Create hard cell command, Node A's 6top sublayer creates a Hard Cell Reservation Request, including a ScheduleIE. The ScheduleIE indicates which specific cells with a neighbor (Node B) to be added. The cells specified in the ScheduleIE SHOULD be added in local schedule of Node A while the Hard Cell Reserve Request is sent to Node B. When receiving the Hard Cell Reserve Request, the cells specified in the ScheduleIE SHOULD be added in the local schedule of Node B.

2.6.6. Deleting hard cells

The upper layer instructs 6top to delete one or more hard cells by calling the Delete hard cell command.

When receiving a Delete hard cell command, Node A's 6top sublayer creates a Hard Cell Remove Request, including a ScheduleIE. The ScheduleIE indicates which specific cells with a neighbor (Node B) to be removed. The cells specified in the ScheduleIE SHOULD be removed from local schedule of Node A while the Hard Cell Remove Request is sent to Node B. When receiving the Hard Cell Remove Request, the cells specified in the ScheduleIE SHOULD be removed from the local schedule of Node B.

2.7. Statistics

The 6top Statistics Function (SF) is responsible for collecting statistics, which it can provide to an upper layer and the Monitoring Function (Section 2.8).

2.7.1. Statistics Metrics

6top is in charge of keeping statistics from a set of metrics gathered from the behavior of the TSCH layer.

The statistics data related to node states and cell metrics SHOULD be provided to upper layer for management, e.g., for RPL to calculate the node's Rank or for GMPLS to the required bandwidth is met. The specific algorithm to generate the statistics is out of scope.

However, the statistics component SHOULD include the following metrics:

1. LinkThroughput: associated with a link, Node A->Node B. For example, LinkThroughput can be calculated with:
$$\text{SUM}(\text{NumOfCell}(i) * \text{NumOfBytePerPacket}) / (\text{FrameLen}(i) * \text{SlotDuration})$$
where NumOfCell(i) is the total number of cells from Node A to Node B in Slotframe-i, FrameLen(i) is the length of Slotframe-i. The unit is Byte/second.
2. Latency: associated with a link, Node A->Node B. For example, latency can be expressed as Minimum and Maximum Latency. Minimum Latency = Min(MinNumOfSlot(i), i=1..) * SlotDuration and Maximum Latency = Max(MaxNumOfSlot(i), i=1..) * SlotDuration where, MinNumOfSlot(i) and MaxNumOfSlot(i) are the minimum or maximum number of timeslots between two dedicated cells from Node A to Node B in Slotframe-i, respectively.
3. LinkQuality. For example, average LQI, ETX;
4. TafficLoad. For example, Queue Full Rate, Queue Empty Rate;
5. NodeEnergy. For example, $E_E = E_{\text{bat}} / [E_0 (T-t)/T]$.

2.7.2. Statistics Configuration

The Statistics Function SHOULD be configurable. The configuration parameters SHOULD include:

LinkQualityStatisticsEn

TafficLoadStatisticsEn

DeviceStatisticsEn

6top statistics function is enabled/disabled and configured by the commands defined in Section 2.4.4

2.8. Monitoring

The 6top Monitoring Function (MF) is responsible for monitoring cell quality, traffic load, and issuing soft cell Maintenance commands, or Create/Delete soft cell commands. The data provided by the Statistics Function MAY be used as an input of MF in taking a monitoring decision.

2.8.1. Monitor Configuration

Monitoring Function SHOULD be configurable. The configuration parameters SHOULD include:

MaintainCellEn.

CreateDeleteCellEn.

QosLevel. QosLevel SHOULD associate with specific neighbor address. QosLevel MAY reflect the latency constraint, cell quality constraint, and so on. The value of QosLevel works as the bandwidth redundancy coefficient.

The 6top monitoring function is enabled/disabled and configured by the commands defined in Section 2.4.3

2.8.2. Actuation

The cell quality statistics MAY be used to generate soft a cell Maintenance command, which triggers a soft cell Maintenance procedure (see Section 2.6.4). The traffic load statistics MAY be used to generate internal Create (resp. Delete) soft cell commands, which triggers a soft cell Reservation (resp. Remove) process (see Section 2.6.2 and Section 2.6.3).

The policy to generate the soft cell Maintenance command and the policy to generate Create/Delete soft cell commands is out of scope.

The policy to generate Create/Delete soft cell commands MAY take QosLevel into account. For example, there are two slotframes existing, Slotframe-1 consists of 32 timeslots, Slotframe-2 consists of 96 timeslots; timeslot duration is 10ms; QosLevel=1.5. If, from the traffic load statistics, MF determines that 2 packet/second SHOULD be added, then the MF generates a Create soft cell command, where FrameID=2, NumCell=3.

2.9. Label Switching

Label Switching Function (LS) in 6top is responsible for maintaining the mapping of input cells and output cells in the same track in a particular node. By keeping that mapping, layer 3 routing can be avoided as packets are forwarded by the 6top sublayer according to the input cells they were received on. The selected output cell is one of the cells that forward the packet to the subsequent hop in the track. As cells can be grouped in bundles, 6top can maintain mappings from input bundles to output bundles and provide a policy to select the output cell according to the input cell.

3. Using 6top

This part describes how 6top gives support to specific upper layers.

3.1. RPL on 6top

6top provides a set of functionalities so higher layers can obtain information about the status of the network and take advantage of the slotted structure to improve metric calculation and objective function optimization. The following sections describe how RPL can make use of 6top sublayer.

In order to optimize the combination of RPL and TSCH, 6top provides specific support to RPL in the following aspects:

- RPL Neighbor Discovery and Parent Selection

- RPL Rank Computation

- RPL Control Messages Broadcast

- QoS

3.1.1. Support to Neighbor Discovery and Parent Selection

The Section 2.4.7 defines a set of commands so the neighbor table can be managed and queried by RPL. An entry to the neighbor table is inserted whenever an EBs is received at L2. The EB causes the 6top sublayer to create an entry to the neighbors table. A neighbor table entry contains a set of statistics with respect to that specific neighbor such as the ASN when the last packet has been received from that neighbor, a set of cell quality metrics (RSSI, LQI), number of packets sent to it or number of packets received from it amongst others. 6top updates that table upon sending or reception of a packet from/to a neighbor. RPL can query at any time the neighbor table to retrieve information about a particular neighbor. This information can be used to compute the routing objective function as for example the Zero Objective function as described in [I-D.vilajosana-6tisch-minimal]. Parent selection can also be driven by the information contained on the neighbor table as well as complemented with the cells statistics defined in Section 2.4.4 and Section 2.7.

6top enables RPL to configure EB periodicity. By controlling the EBs periodicity, RPL can configure how network dynamism and support to mobility are addressed, as more frequent beacons the more prone to cope with mobility. Section 2.4.5 enables to configure how the network is formed and EBs periodicity.

RPL MAY want to select the policy to determine the time source neighbor, this can be interesting when time source neighbors can be aligned to the routing topology, i.e., the selected time source neighbor can be the node's favorite parent in a specific DODAG. Section 2.4.6 describes the 6top command to set up the desired policy. The policy is selected by RPL and enforced by the 6top sublayer.

The rule for 6top to select and maintain time source neighbors is as follows:

The time source neighbor of a node SHOULD be a member of the node's neighbor set.

Time source neighbors SHOULD be the neighbors which have a relatively lower join priority in the neighbor set. A lower join priority indicates that the neighbor is closer to the TSCH PAN coordinator.

The link between a node and one of its time source neighbors SHOULD be a good link quality.

3.1.2. Support of Rank Computation

The RPL objective function is computed using a set of metrics. The [I-D.vilajosana-6tisch-minimal] defines how Zero Objective Function is used to configure the rank and metrics used from 6top statistics. The specific metrics, and how the objective function is calculated are out of scope. However, 6top builds a set of functionalities to provide more accurate statistics of the underlying layer so the objective function can be accommodated to the nature of a TSCH MAC layer.

6top provides statistics for rank computation as described in Section 2.4.4 and Section 2.7. The function used to compute the rank based on those statistics is out of scope. However, the provided metrics are aligned to the behavior of the TSCH MAC layer.

3.1.3. Support of Control Messages Broadcast

In RPL, some control messages, e.g., DIO in storing mode, need to be broadcast to all neighbor nodes. The broadcast channel requirement has to be addressed by 6top by configuring TSCH to provide such a channel.

In order to decouple the upper (RPL) layer from TSCH, instead of carrying DIO messages in Enhance Beacons, 6top introduces a mechanism to establish broadcast cells.

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

To support DIO and DAO broadcasts, 6top uses the payload of a Data Packet to carry the DIO or DAO. The message is inserted into the queue associated with the cells which LinkType is set to ADVERTISING. Then, taking advantage of the broadcast cell feature established with FrameAndLinkIE as described above, the data packet with DIO or DAO in the payload can be received by neighbors, which enforces to the maintenance of DODAG.

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

3.1.4. Support for QoS

The TSCH MAC layer is decoupled from the upper layer, and the interaction between the upper layer and TSCH is asynchronous. This means that the MAC layer executes a schedule and checks at each timeslot according to the type of cell whether there is something to send or receive. If that is the case the packet is transmitted and the MAC layer continues its operation. When an upper layer sends a packet, this packet is pushed into a queue waiting to the MAC layer to read it and send it in a particular timeslot according to its destination and priority (Section 2.3). 6top provides a set of queue management operations which enable upper layers to create different queues and determine their priorities. This allows different classes of traffic to be handled by the routing layer, i.e. inserting a packet to a specific queue according to its priority.

A 6top implement MUST provide at least a Broadcast Queue, a Transmit Queue, and a Receive Queue. RPL can configure the queues with Internal Queueing Command (Section 2.4.8.1). The Broadcast Queue is associated with cells with LinkType=ADVERTISING in sender's schedule, and LinkOption="Receive" and "Timekeeping" in all neighbors'

schedule. This indicates that the cells can be used as broadcast cells from the sender to its neighbors. A Transmit Queue is associated with the dedicated Transmit cells or Shared Cells. RPL can benefit from having different priority queues to improve latency or provide integrated services with different priorities, i.e. different traffic classes.

Data Communication Commands (Section 2.4.10) can be used to send control messages and data messages. The operation is used to insert a message to an specific queue.

For example a suitable configuration can include two Broadcast Queues with priority High and Low, respectively; three Transmit Queues, with priority High, Mid, and Low, respectively; and one Receive Queue.

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

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

6top conducts a QoS policy, which is out of scope. Here is an example. Packets in higher priority queue MUST be sent out before the packets in lower priority queue. Then, when there is an available broadcast/unicast cell, 6top checks the broadcast/unicast queue with higher priority first, if there is a packet, then feeds it to TSCH at the cell, otherwise it checks broadcast/unicast queue with lower priority further. 6top repeats the process, until it finds a broadcast/unicast packet to feed to TSCH or finds that all of broadcast/unicast queues are empty.

3.2. GMPLS on 6top

GMPLS is a 2.5 layer service that is used to forward packets based on the concept of generalized labels. Labels are determined by a reservation protocol during the formation of a multi-hop path. As defined by [RFC3471], [RFC3473] and [RFC4606] a generalized label identifies a flow of data through a set of nodes that conform to a multi-hop path. Instead of being written implicitly into a field in each packet, as is the case in MPLS [RFC3031], the generalized label is kept at each node in the form of a table. The table can be used to map input cells to output cells so routing decisions can be taken at that layer.

In order to optimize the combination of GMPLS and TSCH, 6top provides specific support to GMPLS in the following aspects:

Cell Reservation Support

QoS

3.2.1. Cell Reservation Support for GMPLS on 6top

The GMPLS control plane is used to send path reservation requests and reservation confirmations. When reservation confirmations are received, GMPLS needs to configure the underlying MAC layer to provide the required bandwidth. 6top provides a set of commands to deal with bandwidth allocation, i.e., cell allocation. Section 2.4.1 describes the operations that GMPLS layer MAY use for cell configuration. Note that 6top supports different types of reservations: soft cell and hard cell. How the reservation requirements are expressed is out of scope, but 6top is able to handle a reservation done as a specific bandwidth requirement, done through specifying exact cells.

The [I-D.vilajosana-6tisch-minimal] defines a pre-configured schedule that can be used to bootstrap the network. Those cells can be seen as a GMPLS control plane where RPL routes can be formed and Track reservations issued.

GMPLS can also give different priorities to its control plane and data plane. It can for example be interesting to have a higher priority for control messages so the network adapts to new bandwidth requirements quickly. In contrast, data plane messages can be given a higher priority when they need to meet higher throughput or lower latency. 6top provides commands (Section 2.4.8) to manage MAC layer queues and assign different priorities to them.

3.2.2. Supporting QoS

GMPLS can use 6top statistics to determine whether some QoS requirement is met. Metrics defined in Section 2.7 and operations defined in Section 2.4.4 can be used by GMPLS to trigger new bandwidth allocation, or to map different input bundles to output bundles.

4. References

4.1. Normative References

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

4.2. Informative References

- [RFC2205] Braden, B., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.
- [RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, December 1998.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.
- [RFC3036] Andersson, L., Doolan, P., Feldman, N., Fredette, A., and B. Thomas, "LDP Specification", RFC 3036, January 2001.
- [RFC3471] Berger, L., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description", RFC 3471, January 2003.
- [RFC3473] Berger, L., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReSerVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003.
- [RFC3819] Karn, P., Bormann, C., Fairhurst, G., Grossman, D., Ludwig, R., Mahdavi, J., Montenegro, G., Touch, J., and L. Wood, "Advice for Internet Subnetwork Designers", BCP 89, RFC 3819, July 2004.
- [RFC4606] Mannie, E. and D. Papadimitriou, "Generalized Multi-Protocol Label Switching (GMPLS) Extensions for Synchronous Optical Network (SONET) and Synchronous Digital Hierarchy (SDH) Control", RFC 4606, August 2006.
- [RFC4919] Kushalnagar, N., Montenegro, G., and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", RFC 4919, August 2007.
- [RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, September 2007.
- [RFC5548] Dohler, M., Watteyne, T., Winter, T., and D. Barthel, "Routing Requirements for Urban Low-Power and Lossy Networks", RFC 5548, May 2009.

- [RFC5826] Brandt, A., Buron, J., and G. Porcu, "Home Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5826, April 2010.
- [RFC5867] Martocci, J., De Mil, P., Riou, N., and W. Vermeylen, "Building Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5867, June 2010.
- [RFC5673] Pister, K., Thubert, P., Dwars, S., and T. Phinney, "Industrial Routing Requirements in Low-Power and Lossy Networks", RFC 5673, October 2009.
- [RFC6282] Hui, J. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, September 2011.
- [RFC6550] Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, March 2012.
- [RFC6568] Kim, E., Kaspar, D., and JP. Vasseur, "Design and Application Spaces for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6568, April 2012.
- [RFC6606] Kim, E., Kaspar, D., Gomez, C., and C. Bormann, "Problem Statement and Requirements for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing", RFC 6606, May 2012.
- [RFC6755] Campbell, B. and H. Tschofenig, "An IETF URN Sub-Namespace for OAuth", RFC 6755, October 2012.
- [I-D.watteyne-6tsch-tsch-lln-context]
Watteyne, T., Palattella, M., and L. Grieco, "Using IEEE802.15.4e TSCH in an LLN context: Overview, Problem Statement and Goals", draft-watteyne-6tsch-tsch-lln-context-02 (work in progress), May 2013.
- [I-D.thubert-6tisch-architecture]
Thubert, P., Assimiti, R., and T. Watteyne, "An Architecture for IPv6 over the TSCH mode of IEEE IEEE802.15.4e", draft-thubert-6tisch-architecture-00 (work in progress), October 2013.
- [I-D.palattella-6tisch-terminology]
Palattella, M., Thubert, P., Watteyne, T., and Q. Wang, "Terminology in IPv6 over the TSCH mode of IEEE

802.15.4e", draft-palattella-6tisch-terminology-00 (work in progress), October 2013.

[I-D.vilajosana-6tisch-minimal]

Vilajosana, X. and K. Pister, "Minimal 6TiSCH Configuration", draft-vilajosana-6tisch-minimal-00 (work in progress), October 2013.

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

Chasko, S., Das, S., Lopez, R., Ohba, Y., Thubert, P., and A. Yegin, "Security Framework and Key Management Protocol Requirements for 6TSCH", draft-ohba-6tsch-security-01 (work in progress), July 2013.

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

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

[I-D.tsao-roll-security-framework]

Tsao, T., Alexander, R., Daza, V., and A. Lozano, "A Security Framework for Routing over Low Power and Lossy Networks", draft-tsao-roll-security-framework-02 (work in progress), March 2010.

[I-D.thubert-roll-asymmlink]

Thubert, P., "RPL adaptation for asymmetrical links", draft-thubert-roll-asymmlink-02 (work in progress), December 2011.

[I-D.ietf-roll-terminology]

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

[I-D.ietf-roll-p2p-rpl]

Goyal, M., Baccelli, E., Philipp, M., Brandt, A., and J. Martocci, "Reactive Discovery of Point-to-Point Routes in Low Power and Lossy Networks", draft-ietf-roll-p2p-rpl-17 (work in progress), March 2013.

[I-D.ietf-roll-trickle-mcast]

Hui, J. and R. Kelsey, "Multicast Protocol for Low power and Lossy Networks (MPL)", draft-ietf-roll-trickle-mcast-05 (work in progress), August 2013.

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

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

[I-D.sarikaya-core-sbootstrapping]

Sarikaya, B., Ohba, Y., Moskowitz, R., Cao, Z., and R. Cragie, "Security Bootstrapping Solution for Resource-Constrained Devices", draft-sarikaya-core-sbootstrapping-04 (work in progress), April 2012.

[I-D.gilger-smart-object-security-workshop]

Gilger, J. and H. Tschofenig, "Report from the 'Smart Object Security Workshop', 23rd March 2012, Paris, France", draft-gilger-smart-object-security-workshop-00 (work in progress), October 2012.

[I-D.phinney-roll-rpl-industrial-applicability]

Phinney, T., Thubert, P., and R. Assimiti, "RPL applicability in industrial networks", draft-phinney-roll-rpl-industrial-applicability-02 (work in progress), February 2013.

[I-D.ietf-core-coap]

Shelby, Z., Hartke, K., and C. Bormann, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-18 (work in progress), June 2013.

4.3. External Informative References

[IEEE802154e]

IEEE standard for Information Technology, "IEEE std. 802.15.4e, Part. 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer", April 2012.

[IEEE802154]

IEEE standard for Information Technology, "IEEE std. 802.15.4, Part. 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks", June 2011.

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

[label-switching-154e]

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

Authors' Addresses

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

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

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

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

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

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

6TiSCH
Internet-Draft
Intended status: Informational
Expires: April 23, 2014

T. Watteyne, Ed.
Linear Technology
MR. Palattella
University of Luxembourg
LA. Grieco
Politecnico di Bari
October 20, 2013

Using IEEE802.15.4e TSCH in an LLN context:
Overview, Problem Statement and Goals
draft-watteyne-6tisch-tsch-00

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on April 23, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. TSCH in the LLN Context	4
3. Problems and Goals	5
3.1. Network Formation	6
3.2. Network Maintenance	6
3.3. Multi-Hop Topology	7
3.4. Routing and Timing Parents	7
3.5. Resource Management	7
3.6. Dataflow Control	8
3.7. Deterministic Behavior	8
3.8. Scheduling Mechanisms	8
3.9. Secure Communication	9
4. Acknowledgements	9
5. References	9
5.1. Normative References	9
5.2. Informative References	9
5.3. External Informative References	12
Appendix A. TSCH Protocol Highlights	14
A.1. Timeslots	14
A.2. Slotframes	15
A.3. Node TSCH Schedule	15
A.4. Cells and Bundles	15
A.5. Dedicated vs. Shared Cells	16
A.6. Absolute Slot Number	16
A.7. Channel Hopping	16
A.8. Time Synchronization	17
A.9. Power Consumption	18
A.10. Network TSCH Schedule	18
A.11. Join Process	18
A.12. Information Elements	19
A.13. Extensibility	19
Appendix B. TSCH Gotchas	19
B.1. Collision Free Communication	19
B.2. Multi-Channel vs. Channel Hopping	19
B.3. Cost of (continuous) Synchronization	20
B.4. Topology Stability	20
B.5. Multiple Concurrent Slotframes	20
Authors' Addresses	21

1. Introduction

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

This document describes the main issues arising from the adoption of the IEEE802.15.4e TSCH in the LLN context, following the terminology defined in [I-D.palattella-6tisch-terminology].

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

IEEE802.15.4e can be seen as the latest generation of ultra-lower power and reliable networking solutions for LLNs. [RFC5673] discusses industrial applications, and highlights the harsh operating conditions as well as the stringent reliability, availability, and security requirements for an LLN to operate in an industrial environment. Commercial networking solutions are available today in which motes consume 10's of micro-amps on average [CurrentCalculator] with end-to-end packet delivery ratios over 99.999% [doherty07channel].

IEEE802.15.4e TSCH focuses on the MAC layer only. This clean layering allows for TSCH to fit under an IPv6 enabled protocol stack for LLNs, running 6LoWPAN [RFC6282], RPL [RFC6550] and CoAP [I-D.ietf-core-coap].

Bringing industrial-like performance into the LLN stack developed by the 6LoWPAN, ROLL and CORE working groups opens up new application domains for these networks. Sensors deployed in smart cities [RFC5548] will be able to be installed for years without needing battery replacement. "Umbrella" networks will interconnect smart elements from different entities in smart buildings [RFC5867]. Peel-and-stick switches will obsolete the need for costly conduits for lighting solutions in smart homes [RFC5826].

While [IEEE802154e] defines the mechanisms for a TSCH mote to communicate, it does not define the policies to build and maintain the communication schedule, match that schedule to the multi-hop paths maintained by RPL, adapt the resources allocated between neighbor nodes to the data traffic flows, enforce a differentiated treatment for data generated at the application layer and signalling

messages needed by 6LoWPAN and RPL to discover neighbors, react to topology changes, self-configure IP addresses, or manage keying material.

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

2. TSCH in the LLN Context

In many cases, to map the services required by the IP layer to the services provided by the link layer, an adaptation layer is used [palattella12standardized]. The 6LoWPAN working group started working in 2007 on specifications for transmitting IPv6 packets over IEEE802.15.4 networks [RFC4919]. Typically, low-power WPANs are characterized by small packet sizes, support for addresses with different lengths, low bandwidth, star and mesh topologies, battery powered devices, low cost, large number of devices, unknown node positions, high unreliability, and periods during which communication interfaces are turned off to save energy. Given these features, it is clear that the adoption of IPv6 on top of a Low-Power WPAN is not straightforward, but poses strong requirements for the optimization of this adaptation layer. For instance, due to the IPv6 default minimum MTU size (1280 bytes), an un-fragmented IPv6 packet is too large to fit in an IEEE802.15.4 frame. Moreover, the overhead due to the 40-byte long IPv6 header wastes the scarce bandwidth available at the PHY layer [RFC4944]. For these reasons, the 6LoWPAN working group has defined an effective adaptation layer [RFC6568]. Further issues encompass the auto-configuration of IPv6 addresses [RFC2464][RFC6755], the compliance with the recommendation on supporting link-layer subnet broadcast in shared networks [RFC3819], the reduction of routing and management overhead [RFC6606], the adoption of lightweight application protocols (or novel data encoding techniques), and the support for security mechanisms (confidentiality and integrity protection, device bootstrapping, key establishment, and management).

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

Routing issues are challenging for 6LoWPAN, given the low-power and lossy radio-links, the battery-powered nodes, the multi-hop mesh topologies, and the frequent topology changes due to mobility. Successful solutions take into account the specific application requirements, along with IPv6 behavior and 6LoWPAN mechanisms [palattella12standardized]. The ROLL working group has defined RPL in [RFC6550]. RPL can support a wide variety of link layers, including ones that are constrained, potentially lossy, or typically

utilized in conjunction with host or router devices with very limited resources, as in building/home automation [RFC5867][RFC5826], industrial environments [RFC5673], and urban applications [RFC5548]. RPL is able to quickly build up network routes, distribute routing knowledge among nodes, and adapt to a changing topology. In a typical setting, nodes are connected through multi-hop paths to a small set of root devices, which are usually responsible for data collection and coordination. For each of them, a Destination Oriented Directed Acyclic Graph (DODAG) is created by accounting for link costs, node attributes/status information, and an Objective Function, which maps the optimization requirements of the target scenario. The topology is set up based on a Rank metric, which encodes the distance of each node with respect to its reference root, as specified by the Objective Function. Regardless of the way it is computed, the Rank monotonically decreases along the DODAG towards the destination, building a gradient. RPL encompasses different kinds of traffic and signalling information. Multipoint-to-Point (MP2P) is the dominant traffic in LLN applications. Data is routed towards nodes with some application relevance, such as the LLN gateway to the larger Internet, or to the core of private IP networks. In general, these destinations are the DODAG roots and act as data collection points for distributed monitoring applications. Point-to-Multipoint (P2MP) data streams are used for actuation purposes, where messages are sent from DODAG roots to destination nodes. Point-to-Point (P2P) traffic allows communication between two devices belonging to the same LLN, such as a sensor and an actuator. A packet flows from the source to the common ancestor of those two communicating devices, then downward towards the destination. RPL therefore has to discover both upward routes (i.e. from nodes to DODAG roots) in order to enable MP2P and P2P flows, and downward routes (i.e. from DODAG roots to nodes) to support P2MP and P2P traffic.

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

Several open-source initiatives have emerged around TSCH. The OpenWSN project [OpenWSN][OpenWSNETT] is an open-source implementation of a standards-based protocol stack, which aims at evaluating the applicability of TSCH to different applications. This implementation was used as the foundation for an IP for Smart Objects Alliance (IPSO) [IPSO] interoperability event in 2011. In the absence of a standardized scheduling mechanism for TSCH, a "slotted Aloha" schedule was used.

3. Problems and Goals

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

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

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

3.1. Network Formation

6TiSCH needs to control the way the network is formed, including how new motes join, and how already joined motes advertise the presence of the network. 6TiSCH needs to:

1. Define the Information Elements to include in the Enhanced Beacons advertising the presence of the network.
2. For a new mote, define rules to process and filter received Enhanced Beacons.
3. Define the joining procedure. This includes a mechanism to assign a unique 16-bit address to a mote, and the management of initial keying material.
4. Define a mechanism to secure the joining process and the subsequent optional process of scheduling more communication links.

3.2. Network Maintenance

Once a network is formed, 6TiSCH needs to maintain the network's health, allowing for motes to stay synchronized. 6TiSCH needs to:

1. Manage each mote's time source neighbor.
2. Define a mechanism for a mote to update the join priority it announces in its Enhanced Beacon.

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

3.3. Multi-Hop Topology

RPL, given a weighted connectivity graph, determines multi-hop routes. 6TiSCH needs to:

1. Define a mechanism to gather topological information, which it can then feed to RPL.
2. Ensure that the TSCH schedule contains links along the multi-hop routes identified by RPL.
3. Where applicable, maintain independent sets of links to transport independent flows of data.

3.4. Routing and Timing Parents

At all times, a TSCH mote needs to have a time source neighbor it can synchronize to. 6TiSCH therefore needs to assign a time source neighbor to allow for correct operation of the TSCH network. A time source neighbors could, or not, be taken from the RPL routing parent set.

3.5. Resource Management

A link in a TSCH schedule is a "unit" of resource. The number of links to assign between neighbor motes needs to be appropriate for the size of the traffic flow. 6TiSCH needs to:

1. Define rules on when to create or delete a slotframe.
2. Define rules to determine the length of a slotframe, and the trigger to modify the length of a slotframe.
3. Define rules on when to add or delete links in a particular slotframe.
4. Define a mechanism for neighbor nodes to exchange information about their schedule and, if applicable, negotiate the addition/deletion of links.
5. Allow for an entity (e.g., a set of devices, a distributed protocol, a PCE, etc.) to take control of the schedule.

6. Define a set of metrics to evaluate the trade-off between latency, bandwidth and energy consumption achieved by a particular schedule.

3.6. Dataflow Control

TSCH defines mechanisms for a mote to signal it cannot accept an incoming packet. It does not, however, define the policy which determines when to stop accepting packets. 6TiSCH needs to:

1. Define a queueing policy for incoming and outgoing packets.
2. Manage the buffer space, and indicate to TSCH when to stop accepting incoming packets.
3. Handle transmissions that have failed. A transmission is declared failed when TSCH has retransmitted the packet multiple times, without receiving an acknowledgement. This covers both dedicated and shared links.

3.7. Deterministic Behavior

As highlighted in [RFC5673], in some applications, data is generated periodically and has a well understood data bandwidth requirement, which is deterministic and predictable. 6TiSCH needs to:

1. Ensure timely delivery of such data.
2. Provide a mechanism for such deterministic flows to coexist with bursty or infrequent traffic flows of different priorities.

3.8. Scheduling Mechanisms

Several scheduling mechanisms can be envisioned, and possibly coexist in the same network. For example, [I-D.phinney-roll-rpl-industrial-applicability] describe how the allocation of bandwidth can be optimized by an external Path Computation Element (PCE). Alternatively, two neighbor nodes can adapt the number of cells autonomously by monitoring the amount of traffic, and negotiating the allocation to extra cell when needed. This mechanism can be used to establish multi-hop paths in a fashion similar to RSVP. 6TiSCH needs to:

1. Provide a mechanism for two 6TiSCH devices to negotiate the allocation and deallocation of cells between them.
2. Provide a mechanism for device to monitor and manage the 6TiSCH capabilities of a node several hops away.

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

3.9. Secure Communication

Given some keying material, TSCH defines mechanisms to encrypt and authenticate MAC frames. It does not define how this keying material is generated. 6TiSCH needs to:

1. Define the keying material and authentication mechanism needed by a new mote to join an existing network.
2. Define a mechanism to allow for the secure transfer of application data between neighbor motes.
3. Define a mechanism to allow for the secure transfer of signalling data between motes and 6TiSCH.

4. Acknowledgements

Special thanks to Jonathan Simon for his review and valuable comments. Thanks to the IoT6 European Project (STREP) of the 7th Framework Program (Grant 288445).

5. References

5.1. Normative References

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

5.2. Informative References

- [RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, December 1998.
- [RFC3819] Karn, P., Bormann, C., Fairhurst, G., Grossman, D., Ludwig, R., Mahdavi, J., Montenegro, G., Touch, J., and L. Wood, "Advice for Internet Subnetwork Designers", BCP 89, RFC 3819, July 2004.
- [RFC4919] Kushalnagar, N., Montenegro, G., and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", RFC 4919, August 2007.

- [RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, September 2007.
- [RFC5548] Dohler, M., Watteyne, T., Winter, T., and D. Barthel, "Routing Requirements for Urban Low-Power and Lossy Networks", RFC 5548, May 2009.
- [RFC5826] Brandt, A., Buron, J., and G. Porcu, "Home Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5826, April 2010.
- [RFC5867] Martocci, J., De Mil, P., Riou, N., and W. Vermeulen, "Building Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5867, June 2010.
- [RFC5673] Pister, K., Thubert, P., Dwars, S., and T. Phinney, "Industrial Routing Requirements in Low-Power and Lossy Networks", RFC 5673, October 2009.
- [RFC6282] Hui, J. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, September 2011.
- [RFC6550] Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, March 2012.
- [RFC6568] Kim, E., Kaspar, D., and JP. Vasseur, "Design and Application Spaces for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6568, April 2012.
- [RFC6606] Kim, E., Kaspar, D., Gomez, C., and C. Bormann, "Problem Statement and Requirements for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing", RFC 6606, May 2012.
- [RFC6755] Campbell, B. and H. Tschofenig, "An IETF URN Sub-Namespace for OAuth", RFC 6755, October 2012.
- [I-D.wang-6tsch-6top]
Wang, Q., Vilajosana, X., and T. Watteyne, "6TSCH Operation Sublayer (6top)", draft-wang-6tsch-6top-00 (work in progress), July 2013.
- [I-D.palattella-6tisch-terminology]

Palattella, M., Thubert, P., Watteyne, T., and Q. Wang, "Terminology in IPv6 over the TSCH mode of IEEE 802.15.4e", draft-palattella-6tisch-terminology-00 (work in progress), October 2013.

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

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

[I-D.tsao-roll-security-framework]

Tsao, T., Alexander, R., Daza, V., and A. Lozano, "A Security Framework for Routing over Low Power and Lossy Networks", draft-tsao-roll-security-framework-02 (work in progress), March 2010.

[I-D.thubert-roll-asymlink]

Thubert, P., "RPL adaptation for asymmetrical links", draft-thubert-roll-asymlink-02 (work in progress), December 2011.

[I-D.ietf-roll-terminology]

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

[I-D.ietf-roll-p2p-rpl]

Goyal, M., Baccelli, E., Philipp, M., Brandt, A., and J. Martocci, "Reactive Discovery of Point-to-Point Routes in Low Power and Lossy Networks", draft-ietf-roll-p2p-rpl-17 (work in progress), March 2013.

[I-D.ietf-roll-trickle-mcast]

Hui, J. and R. Kelsey, "Multicast Protocol for Low power and Lossy Networks (MPL)", draft-ietf-roll-trickle-mcast-05 (work in progress), August 2013.

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

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

[I-D.sarikaya-core-sbootstrapping]

Sarikaya, B., Ohba, Y., Moskowitz, R., Cao, Z., and R. Cragie, "Security Bootstrapping Solution for Resource-Constrained Devices", draft-sarikaya-core-sbootstrapping-04 (work in progress), April 2012.

- [I-D.gilger-smart-object-security-workshop]
Gilger, J. and H. Tschofenig, "Report from the 'Smart Object Security Workshop', 23rd March 2012, Paris, France", draft-gilger-smart-object-security-workshop-00 (work in progress), October 2012.
- [I-D.phinney-roll-rpl-industrial-applicability]
Phinney, T., Thubert, P., and R. Assimiti, "RPL applicability in industrial networks", draft-phinney-roll-rpl-industrial-applicability-02 (work in progress), February 2013.
- [I-D.ietf-core-coap]
Shelby, Z., Hartke, K., and C. Bormann, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-18 (work in progress), June 2013.

5.3. External Informative References

- [IEEE802154e]
IEEE standard for Information Technology, "IEEE std. 802.15.4e, Part. 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer", April 2012.
- [IEEE802154]
IEEE standard for Information Technology, "IEEE std. 802.15.4, Part. 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks", June 2011.
- [OpenWSN] , "Berkeley's OpenWSN Project Homepage", , <<http://www.openwsn.org/>>.
- [OpenWSNETT]
Watteyne, T., Vilajosana, X., Kerkez, B., Chraim, F., Weekly, K., Wang, Q., Glaser, S., and K. Pister, "OpenWSN: a standards-based low-power wireless development environment", Transactions on Emerging Telecommunications Technologies 2012, August 2012, <<http://onlinelibrary.wiley.com/doi/10.1002/ett.2558/abstract>>.
- [IPSO] , "IP for Smart Objects Alliance Homepage", , <<http://www.ipso-alliance.org/>>.
- [CurrentCalculator]
Linear Technology, "Application Note: Using the Current Calculator to Estimate Mote Power", August 2012, <<http://>>

cds.linear.com/docs/en/application-note/Application_Note_-_Using_the_Current_Calculator_to_Estimate_Mote_Power.pdf>.

[doherty07channel]

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

[tinka10decentralized]

Tinka, A., Watteyne, T., and K. Pister, "A Decentralized Scheduling Algorithm for Time Synchronized Channel Hopping", Ad Hoc Networks 2010, 2010, < <http://robotics.eecs.berkeley.edu/~pister/publications/2008/TSMPT%20DSN08.pdf>>.

[watteyne09reliability]

Watteyne, T., Mehta, A., and K. Pister, "Reliability Through Frequency Diversity: Why Channel Hopping Makes Sense", International Conference on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN) 2009, Oct. 2009, <http://www.ietf.org/mail-archive/web/roll/current/pdfa_EzmuDIv3.pdf>.

[kerkez09feasibility]

Kerkez, B., Watteyne, T., and M. Magliocco, "Feasibility analysis of controller design for adaptive channel hopping", International Workshop on Performance Methodologies and Tools for Wireless Sensor Networks (WSNPERF) 2009, Oct. 2009, <http://www-bsac.eecs.berkeley.edu/publications/search/send_publication_pdf2client.php?pubID=1249681245>.

[TASA-PIMRC]

Palattella, MR., Accettura, N., Dohler, M., Grieco, LA., and G. Boggia, "Traffic Aware Scheduling Algorithm for Multi-Hop IEEE 802.15.4e Networks", IEEE PIMRC 2012, Sept. 2012, < <http://www.cttc.es/resources/doc/120531-submitted-tasa-25511.pdf>>.

[TASA-SENSORS]

Palattella, MR., Accettura, N., Dohler, M., Grieco, LA., and G. Boggia, "Traffic-Aware Time-Critical Scheduling In Heavily Duty-Cycled IEEE 802.15.4e For An Industrial IoT", IEEE SENSORS 2012, Oct. 2012, < <http://www.cttc.es/resources/doc/120821-sensors2012-4396981770946977737.pdf>>.

[TASA-WCNC]

Accettura, N., Palattella, MR., Dohler, M., Grieco, LA., and G. Boggia, "Standardized Power-Efficient and Internet-Enabled Communication Stack for Capillary M2M Networks", IEEE WCNC 2012, Apr. 2012, < <http://www.cttc.es/resources/doc/120109-1569521283-submitted-58230.pdf>>.

[palattella12standardized]

Palattella, MR., Accettura, N., Vilajosana, X., Watteyne, T., Grieco, LA., Boggia, G., and M. Dohler, "Standardized Protocol Stack For The Internet Of (Important) Things", IEEE Communications Surveys and Tutorials 2012, Dec. 2012, < <http://www.cttc.es/resources/doc/121025-completestackforiot-clean-4818610916636121981.pdf>>.

[PANA]

Kanda, M., Ohba, Y., Das, S., and S. Chasko, "PANA applicability in constrained environments", Febr. 2012, <<http://www.lix.polytechnique.fr/hipercom/SmartObjectSecurity/papers/MitsuruKanda.pdf>>.

Appendix A. TSCH Protocol Highlights

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

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

A.1. Timeslots

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

The duration of a timeslot is not defined by the standard. With IEEE802.15.4-compliant radios operating in the 2.4GHz frequency band, a maximum-length frame of 127 bytes takes about 4ms to transmit; a shorter ACK takes about 1ms. With a 10ms slot (a typical duration), this leaves 5ms to radio turnaround, packet processing and security operations.

A.2. Slotframes

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

A.3. Node TSCH Schedule

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

Once a mote obtains its schedule, it executes it:

- o For each transmit cell, the mote checks whether there is a packet in the outgoing buffer which matches the neighbor written in the schedule information for that timeslot. If there is none, the mote keeps its radio off for the duration of the timeslot. If there is one, the mote can ask for the neighbor to acknowledge it, in which case it has to listen for the acknowledgement after transmitting.
- o For each receive cell, the mote listens for possible incoming packets. If none is received after some listening period, it shuts down its radio. If a packet is received, addressed to the mote, and passes security checks, the mote can send back an acknowledgement.

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

A.4. Cells and Bundles

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

A single element of the schedule characterized by a slotOffset and channelOffset, and reserved for mote A to transmit to mote B (or for mote B to receive from mote A) within a given slotframe, is called a "scheduled cell".

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

A.5. Dedicated vs. Shared Cells

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

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

A.6. Absolute Slot Number

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

$$\text{ASN} = (k \cdot S + t)$$

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

A.7. Channel Hopping

For each scheduled cell, the schedule specifies a slotOffset and a channelOffset. In a well-built schedule, when mote A has a transmit

cell to mote B on channelOffset 5, mote B has a receive cell from mote A on the same channelOffset. The channelOffset is translated by both nodes into a frequency using the following function:

$$\text{frequency} = F \{(\text{ASN} + \text{channelOffset}) \bmod \text{nFreq}\}$$

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

This results in "channel hopping": even with a static schedule, pairs of neighbors "hop" between the different frequencies when communicating. Channel hopping is a technique known to efficiently combat multi-path fading and external interference.

A.8. Time Synchronization

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

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

TSCH adds timing information in all packets that are exchanged (both data and ACK frames). This means that neighbor motes can resynchronize to one another whenever they exchange data. In detail, in the IEEE 802.15.4e standard two methods are defined for allowing a device to synchronize in a TSCH network: (i) Acknowledgement-Based and (ii) Frame-Based synchronization. In both cases, the receiver calculates the difference in time between the expected time of frame arrival and its actual arrival. In Acknowledgement-Based synchronization, the receiver provides such information to the sender

mote in its acknowledgement. Thus, in this case, it is the sender mote that synchronizes to the clock of the receiver. In Frame-Based synchronization, the receiver uses the computed delta for adjusting its own clock. Therefore, it is the receiver mote that synchronizes to the clock of the sender.

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

A.9. Power Consumption

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

A.10. Network TSCH Schedule

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

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

A.11. Join Process

Motes already part of the network can periodically send Enhanced Beacon (EB) frames to announce the presence to the network. These contain information about the size of the timeslot used in the network, the current ASN, information about the slotframes and timeslots the beaconing mote is listening on, and a 1-byte join priority. Because of the channel hopping nature of TSCH, these EB frames are sent on all frequencies.

A mote wishing to join the network listens for EBs. Using the ASN and the other timing information of the EB, the new mote synchronizes to the network. Using the slotframe and link information from the EB, it knows how to contact the network.

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

A.12. Information Elements

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

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

A.13. Extensibility

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

Appendix B. TSCH Gotchas

This section lists features of TSCH which we believe are important and beneficial to the work of 6TiSCH.

B.1. Collision Free Communication

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

B.2. Multi-Channel vs. Channel Hopping

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

B.3. Cost of (continuous) Synchronization

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

B.4. Topology Stability

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

B.5. Multiple Concurrent Slotframes

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

1. Transmissions take precedence over receptions;

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

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

Authors' Addresses

Thomas Wattheyne (editor)
Linear Technology
30695 Huntwood Avenue
Hayward, CA 94544
USA

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

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

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

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

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