6TiSCH
Internet-Draft
Intended status: Informational
Expires: April 21, 2016
Q. Wang, Ed.
Univ. of Sci. and Tech. Beijing
X. Vilajosana
Universitat Oberta de Catalunya
October 19, 2015

6TiSCH Operation Sublayer (6top)

draft-wang-6tisch-6top-sublayer-03

Abstract

This document defines the 6TiSCH Operation Sublayer (6top), which offers mechanisms for distributed scheduling in 6TiSCH networks. The 6top sublayer is the next higher layer of the IEEE802.15.4e TSCH medium access control layer. The 6top Protocol (6P) defined in this document allows neighbor nodes to add/delete TSCH cells to one another. To be able to match different application requirements, the 6top Scheduling Function (SF) decides when to add/delete cells. The SF is left out of scope, and will be specified in one or more companion documents.

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

Table of Contents

1. Introduction

All communication in a 6TiSCH network is orchestrated by a schedule [RFC7554]. This specification defines the mechanisms offered by the 6TiSCH Operation Sublayer (6top) sublayer. These mechanisms allow a node to communicate with its neighbor node(s) to agree on a TSCH schedule in a distributed manner.

```
       (A)
       / \
      /   \
   (B)-----(C)
    |       |
    |       |
   (D)     (E)
```

Figure 1: A simple 6TiSCH network.

For example, node C in Figure 1 monitors the communication cells to node A it has in its schedule.

- If node C determines the number of frames it is sending to A per unit of time is larger than the capacity offered by the TSCH cells it has scheduled to A, it communicates with node A to add one or more such cells.
- If the traffic is lower than the capacity, node C communicates with node A to delete one or more cells to A.
- Node C might also monitor statistics to determine whether collisions are happening on a particular cell to node A. If this feature is enabled, node C communicates with node A to add a new cell and delete the cell which suffered from collisions. This results, conceptually, in "relocating" the cell which suffered from collisions to a different slotOffset/channelOffset location in the TSCH schedule. The mechanism handling cell relocation is out of the scope of this document.

This results in a distributed schedule management solution.

The mechanisms needed to enable this interaction are defined by the 6TiSCH Operation Sublayer (6top) sublayer, described in Section 2. The 6top Protocol (6P), specified in Section 3, defines the communication between neighbor nodes in this context. The 6top sublayer includes a 6top Scheduling Function (SF) which defines the policy of when to add/delete a cell to a neighbor. Different applications require different SFs, so the SF is left out of scope of this document. One or more SFs will be defined in one or more companion documents. Section 4 provides some guidelines on how to design an SF.

2. 6TiSCH Operation Sublayer (6top)

As depicted in Figure 2, the 6TiSCH Operation Sublayer (6top) sits directly above the IEEE802.15.4e TSCH medium access control layer [IEEE802154e].

```
                          .
     |                    .                      |
     |             next higher layer             |
     +-------------------------------------------+
     |                   6top                    |
     +-------------------------------------------+
     |            IEEE802.15.4e TSCH             |
     |                    .                      |
                          .
```

Figure 2: The 6top sublayer in the protocol stack.

The roles of the 6top sublayer are:

- o Implement and terminate the 6top Protocol (6P), which allows neighbor nodes to communicate to add/delete cells to one another.
- o Run a 6top Scheduling Function (SF) which defines the algorithm to decide when to add/delete cells.
- o Offer a way for a neighbor node to discover which SF is being used.

2.1. Hard/Soft Cells

6top qualifies each cell in the schedule as either "hard" or "soft":

- o a Soft Cell can be read, added, deleted or updated by 6top.
- o a Hard Cell is read-only for 6top.

In the context of this specification, all the cells used by 6top are Soft Cells. Hard cells can be used for example when "hard-coding" a cell (e.g. the 6TiSCH Configuration [I-D.ietf-6tisch-minimal]).

2.2. Using 6top with the Minimal 6TiSCH Configuration

6top MAY be used alongside the Minimal 6TiSCH Configuration [I-D.ietf-6tisch-minimal]. In this case, it is RECOMMENDED to use 2 slotframes, as depicted in Figure 3:

- o Slotframe 0 (SFR0) is used for traffic defined in the Minimal 6TiSCH Configuration. In Figure 3, this slotframe is 5 slots long, but it can be of any length.
- o Slotframe 1 (SFR1) is used by 6top to allocate cells from. In Figure 3, this slotframe is 10 slots long, but it can be of any length.

.

SFR0 SHOULD be of higher priority than SFR1. 6top MAY support further slotframes; how to use more slotframes is out of the scope for this document.

```
         | 0    1    2    3    4 | 0    1    2    3    4  |
         +-----------------------+------------------------+
    SFR0 | EB |    |    |    |    | EB |    |    |    |    |
         |    |    |    |    |    |    |    |    |    |    |
         +-------------------------------------------------+
    SFR1 |    |A->B|    |    |    |    |    |    |B->A|    |
         |    |    |    |    |    |    |    |    |    |    |
         +-------------------------------------------------+
         | 0    1    2    3    4    5    6    7    8    9  |
```

Figure 3: 2-slotframe structure when using 6top alongside the Minimal 6TiSCH Configuration.

3. 6top Protocol (6P)

The 6top Protocol (6P) allows two neighbor nodes to pass information to add/delete cells to their TSCH schedule. This information is carried as IEEE802.15.4 Information Elements (IE) [IEEE802154e] and travels only a single hop.

Conceptually, two neighbor nodes "negotiate" the location of the cells to add/delete. We reuse the topology in Figure 1 to illustrate how the protocol works.

When node A wants to add (resp. delete) 2 cells to node B:

1. Node A sends a message to node B indicating it wants to add (resp. delete) 2 cells to node B to its schedule, and listing 2 or more candidate cells.
2. Node B responds with a message indicating that the operation succeeded, and specifying which cells from the candidate list it added (resp. deleted). This allows node A to add (resp. delete) the same cells to/from its schedule.

Figure 4 is a sequence diagram which illustrates this exchange. Here, node A requests 2 cells to node B. It sends a 6P ADD Request to node be indicatig it wishes to add 2 cells (the "NumCells" value), and specifying a list of 3 candidate cells from which node B can choose (the "CellList" value). Each cell in the CellList is a tuple with the (slotOffset,channelOffset) coordinates of the candidate cell in the TSCH schedule. Node B selects 2 of the 3 cells in the CellList of the 6P ADD Request, and sends a 6P Response back to node A specifying the cells it selected from the specified container (e.g Slotframe, Chunk, etc ...). This allow nodes A and B to add those two cells to their schedule.

```
+----------+                       +----------+
|  Node A  |                       |  Node B  |
+----+-----+                       +-----+----+
     |                                   |
     | 6P ADD Request                    |
     |   NumCells     = 2                |
     |   Container ID = 1                |
     |   CellList     = [(1,2),(2,2),(3,5)]  |
     |---------------------------------------->|
     |                                   |
     |       6P Response                 |
     |         Return Code  = IANA_RC_SUCCESS|
     |         CellList     = [(2,2),(3,5)]  |
     |<----------------------------------------|
     |                                   |
```

Figure 4: Sequence diagram to illustrate the 6P negotiation.

We call "6P Transaction" the action of two neighbor nodes exchanging a 6P Request Message and the corresponding 6P Reply message.

3.1. Message Format

3.1.1. 6top Information Element

The messages exchanges as part of the 6P protocol are carried in a 6top Information Element. The 6top Information Element is a IETF IE with Group ID IANA_6TOP_IE_GROUP_ID. The length of the 6top

Information Element is variable. The content of the 6top Information Element is specified in Section 3.1. TODO: IETF IE specified in Appendix A for now, but to be specified in separate draft in the future.

3.1.2. General Message Format

All 6P messages have the following format:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Ver   | Code  |     SFID      | Other Fields
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Ver (6P Version): The version of the 6P protocol. Only version IANA_6P_VERSION is defined in this document. Future specification might define further version of the 6P protocol.

Code: Command to carry out, or response code. The list of command identifiers and return codes is defined only for version IANA_6P_VERSION in this document.

SFID (6top Scheduling Function Identifier): The identifier of the SF to use to handle this message. The SFID is defined in Section 4.1.

Other Fields: The list of other fields depends on the value of the code field, as detailed below.

3.1.3. 6P Command Identifiers

Figure 5 lists the 6P command identifiers.

Value	Command ID	Description
IANA_CMD_ADD	CMD_ADD	add one or more cells
IANA_CND_DELETE	CMD_DELETE	delete one or more cells
IANA_CMD_COUNT	CMD_COUNT	count scheduled cells
IANA_CMD_LIST	CMD_LIST	list the scheduled cells
IANA_CMD_CLEAR	CMD_CLEAR	clear all cells
TODO-0xf	reserved	

Figure 5: 6P Command Identifiers

3.1.4. 6P Return Codes

Figure 6 lists the 6P Return Codes and their meaning.

Value	Return Code	Description
IANA_RC_SUCCESS	RC_SUCCESS	operation succeeded
IANA_RC_VER_ERR	RC_VER_ERR	unsupported 6P version
IANA_RC_SFID_ERR	RC_SFID_ERR	unsupported SFID
IANA_RC_ERR_BUSY	RC_ERR_BUSY	handling previous request
IANA_RC_RESET	RC_RESET	abort 6P transaction
IANA_RC_ERR	RC_ERR	operation failed
TODO-0xf	reserved	

Figure 6: 6P Return Codes

3.1.5. 6P Cell Format

The 6P Cell is an element which is present in several messages. It is a 4-byte field formatted as:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          slotOffset           |         channelOffset         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

slotOffset: The slot offset of the cell.
channelOffset: The channel offset of the cell.

3.1.6. 6P ADD Request Format

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Ver   |  Code |     SFID      | NumCells      | Container     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| CellList ...
+-+-+-+-+-+-+-
```

Ver: Set to IANA_6P_VERSION.

Code: Set to IANA_CMD_ADD for a 6P ADD Request.
SFID: Identifier of the SF to be used by the receiver to handle the message.
NumCells: The number of additional TX cells the sender wants to schedule to the receiver.
Container: An indication of where in the schedule to take the cells from (which slotframe, which chunk, etc.). This value is an indication to the SF. The meaning of this field depends on the SF, and is hence out of scope of this document.
CellList: A list of 0, 1 or multiple 6P Cells. The format of a 6P Cell is defined in Section 3.1.5

3.1.7. 6P DELETE Request Format

The 6P DELETE Request has the exact same format as the 6P ADD Request, except for the code which is set to IANA_CMD_DELETE.

3.1.8. 6P COUNT Request Format

```
                     1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Ver   |  Code |    SFID       | Container     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Ver: Set to IANA_6P_VERSION.
Code: Set to IANA_CMD_COUNT for a 6P COUNT Request.
SFID: Identifier of the SF to be used by the receiver to handle the message.
Container: An indication of where in the schedule to take the cells from (which slotframe, which chunk, etc.). This value is an indication to the SF. The meaning of this field depends on the SF, and is hence out of scope of this document.

3.1.9. 6P LIST Request Format

The 6P LIST Request has the exact same format as the 6P COUNT Request, except for the code which is set to IANA_CMD_LIST.

3.1.10. 6P CLEAR Request Format

The 6P CLEAR Request has the exact same format as the 6P COUNT Request, except for the code which is set to IANA_CMD_CLEAR.

3.1.11. 6P Response Format

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Ver   |  Code |     SFID      | Other Fields ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Ver: Set to IANA_6P_VERSION.

SFID: Identifier of the SF to be used by the receiver to handle the message.

Code: One of the 6P Return Codes listed in Section 3.1.4.

Other Fields: The fields depends on what command the request is for:

- Response to an ADD, DELETE or LIST command: A list of 0, 1 or multiple 6P Cells. The format of a 6P Cell is defined in Section 3.1.5.
- Response to COUNT command: The number of cells scheduled from the requestor to the receiver by the 6P protocol, encoded as a 2-octet unsigned integer.
- Response to CLEAR command: No other fields are present in the response.

3.2. Protocol Behavior

For illustration, we assume we use the topology in Figure 1, and that node A negotiates to add/delete cells to node B.

3.2.1. Version Checking

All messages contain a Version field. If multiple Versions of the 6P protocol have been defined (in future specifications for Version values different than IANA_6P_VERSION), a node MAY implement multiple protocol versions at the same time. When receiving a 6P message with a Version number it does not implement, a node MUST reply with a 6P Response and a return code of IANA_RC_VER_ERR. The Version field in the 6P Response MUST be the same as the Version field in the corresponding 6P Request.

3.2.2. SFID Checking

All messages contain a SFID field. If multiple SFs has been defined, a node MAY support multiple SFs at the same time. When receiving a 6P message with an unsupported SFID, a node MUST reply with a 6P Response and a return code of IANA_RC_SFID_ERR. The Version field in the 6P Response MUST be the same as the Version field in the corresponding 6P Request.

3.2.3. Concurrent 6P Transactions

Only a single 6P Transaction between two neighbors, in a given direction, can take place at the same time. That is, a node MUST NOT issue a new 6P Request to a given neighbor before having received the 6P Response for a previous request to that neighbor. The only exception to this rule is when the previous 6P Transaction has timed out. If a node receives a 6P Request from a given neighbor before having sent the 6P Response to the previous 6P Request from that neighbor, it MUST send back a 6P Response with a return code of IANA_RC_ERR.

A node MAY support concurrent 6P Transactions from different neighbors. In this case, in Figure 1, node C can have a different ongoing 6P Transaction with nodes B and E. In case a node does not have enough resources to handle concurrent 6P Transactions from different neighbors, when it receives a 6P Request from a neighbor while already handling a different request from a different neighbor, it MUST reply to that second request with a 6P Response with return code IANA_RC_BUSY.

3.2.4. Timeout

A timeout happens when the node sending the 6P Request has not received the 6P Response. The value of the timeout is coupled with how the cells between the nodes are scheduled. The SF determines the value of the timeout. The value of the timeout is out of scope of this document.

3.2.5. Adding cells

We assume the topology in Figure 1 where the SF on node C decides to add NumCell cells to node A.

Node C's SF selects NumCandidate>=NumCell cells from its schedule as candidate transmit cells to node A. NumCandidate MUST be larger or equal to NumCell. How many cells it selects (NumCandidate) and how that selection is done is specified in the SF and out of scope of this document. Node C sends a 6P ADD Request to node A which contains the value of NumCells and the NumCandidate cells in the CellList.

Upon receiving the request, node A's SF verifies which of the cells in the CellList it can add as receive cells from node C in its own schedule. How that selection is done is specified in the SF and out of scope of this document. That verification can succeed (NumCell cells from the CellList can be used), fail (none of the cells from the CellList can be used) or partially succeed (less than NumCell

cells from the CellList can be used). In all cases, node A MUST send a 6P Response with return code set to IANA_RC_SUCCESS, and which specifies the list of cells that were scheduled as receive cells from C. That can contain 0 elements (when the verification failed), NumCell elements (succeeded) or between 0 and NumCell elements (partially succeeded).

Upon receiving the response, node C adds the cells specified in the CellList as transmit cells to node A.

3.2.6. Aborting a 6P Transaction

In case the receiver of a 6top request fails during a 6P Transaction and is unable to complete it, it SHOULD reply to that request with a 6P Response with return code IANA_RC_ERR_RESET. Upon receiving this 6top reply, the initiator of the 6P Transaction MUST consider the 6P Transaction as failed.

3.2.7. Deleting cells

The behavior for deleting cells is equivalent to that of adding cells except that:

- o The nodes delete the cells they agree upon rather than adding them.
- o All cells in the CellList MUST be already scheduled between the two nodes.
- o If the CellList in the 6P Request is empty, the SF on the receiving node is free to delete any cell from the sender.
- o The CellList MUST either be equal, contain exactly NumCell cells, or more than NumCell cells. The case where the CellList is not empty but contains less than NumCell cells is not supported.

3.2.8. Handling error responses

A return code with a name starting with "RC_ERR" as in Figure 6 indicates an error. When a node receives a 6P Response with such an error, it MUST consider the 6P Transaction failed. In particular, if this was a response to a 6P ADD/DELETE Request, the node MUST NOT add/delete any of the cells involved in this 6P Transaction. Similarly, a node sending a 6P Response with an "RC_ERR" return code MUST NOT add/delete any cells as part of that 6P Transaction. The SF defines what to do after an error has occurred. Defining what to do after an error has occurred is out of scope of this document.

3.3. Security

6P messages are secured through link-layer security. When link-layer security is enabled, the 6P messages MUST be secured. This is possible because 6P messages are carried as Payload IE.

4. Guidelines for 6top Scheduling Functions (SF)

4.1. SF Identifier (SFID)

Each SF has an identifier. The identifier is encoded as a 1-byte field. The identifier space is divided in the following ranges.

Range	Meaning
0x00	reserved
0x01-0xef	managed
0xf0-0xfe	unmanaged
0xff	reserved

Figure 7: SFID range.

SF identifiers in the managed space MUST be managed by IANA.

4.2. Requirements for an SF

The specification for an SF

- o MUST specify an identifier for that SF.
- o SHOULD clearly state the application domain the SF is created for.
- o MUST specify the rule for a node to decide when to add/delete one or more cells to a neighbor.
- o MUST specify the rule for a Transaction source to select cells to add to the CellList field in the 6P ADD Request.
- o MUST specify the rule for a Transaction destination to select cells from CellList to add to its schedule.
- o MUST specify a value for the 6P Timeout, or a rule to calculate it.
- o MUST specify a meaning for the "Container" field in the 6P ADD Request.
- o MUST specify the behavior of a node when it boots.
- o MUST specify what to do after an error has occurred (either the node sent a 6P Response with an error code, or received one).

- o SHOULD contain examples which highlight normal and error scenarios.
- o SHOULD contain a list of current implementations, at least during the I-D state of the document, per [RFC6982].
- o SHOULD contain a performance evaluation of the scheme, possibly through references to external documents.

4.3. Recommended Structure of an SF Specification

The following section structure for a SF document is RECOMMENDED:

- o Introduction
- o Scheduling Function Identifier
- o Rules for Adding/Deleting Cells
- o Rules for CellList
- o 6P Timeout Value
- o Meaning of Container Field
- o Node Behavior at Boot
- o 6P Error Handling
- o Examples
- o Implementation Status
- o Security Considerations
- o IANA Considerations

5. Implementation Status

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC6982]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC6982], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

OpenWSN: This specification is implemented in the OpenWSN project [OpenWSN]. The authors of this document are collaborating with

the OpenWSN community to gather feedback about the status and performance of the protocols described in this document. Results from that discussion will appear in this section in future revision of this specification.

6. Security Considerations

TODO: analyze risks

6P messages are carried inside IEEE802.15.4 Payload Information Elements (IEs). Those Payload IEs are encrypted and authenticated at the link layer through CCM*. 6P benefits from the same level of security as any other Payload IE. The 6P protocol does not define its own security mechanisms. A key management solution is out of scope for this document. The 6P protocol will benefit for the key management solution used in the network.

7. IANA Consideration

- TODO: IANA_6TOP_IE_GROUP_ID
- TODO: IANA_6P_VERSION
- TODO: IANA_CMD_ADD
- TODO: IANA_CMD_DELETE
- TODO: IANA_RC_SUCCESS
- TODO: IANA_RC_VER_ERR
- TODO: IANA_RC_ERR

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[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.

8.2. Informative References

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

[RFC6982] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", RFC 6982, DOI 10.17487/RFC6982, July 2013, <http://www.rfc-editor.org/info/rfc6982>.

[I-D.ietf-6tisch-minimal] Vilajosana, X. and K. Pister, "Minimal 6TiSCH Configuration", draft-ietf-6tisch-minimal-12 (work in progress), September 2015.

[I-D.ietf-6tisch-terminology] Palattella, M., Thubert, P., Watteyne, T., and Q. Wang, "Terminology in IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-terminology-05 (work in progress), July 2015.

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

Appendix A. [TEMPORARY] IETF IE

This section contains a proposal for the specification of an IETF IE. If this proposal is supported by the 6TiSCH WG, the authors of this draft recommend for the specification of the IETF IE to be its own draft, possibly developed in the 6TiSCH WG. The reason for having it a separated document is that the scope of the IETF IE is wider that the 6P protocol defined in this document.

The IETF IE is a IEEE802.15.4 Payload Information Element with the Group ID set to IANA_6TOP_IE_GROUP_ID. The value of IANA_6TOP_IE_GROUP_ID is defined by the IEEE, communicated to the IETF, and noted by IANA. The format of the IETF IE is exactly the same as the format of an MLME Information Element, as specified in [IEEE802154e], Section 5.2.4.5. The difference is that the space of Sub-IDs is managed by the IETF/IANA.

Appendix B. [TEMPORARY] IEEE Liaison Considerations

If the specification described in this document is supported by the 6TiSCH WG, the authors of this document ask the 6TiSCH WG chairs to liaise with the IEEE to request a Payload Information Element Group ID to be assigned to the IETF (Group ID IANA_6TOP_IE_GROUP_ID described in Appendix A).

Appendix C. [TEMPORARY] Terms for the Terminology Draft

Terms introduced by this document, and which needs to be added to [I-D.ietf-6tisch-terminology]:

6top: The "6TiSCH Operation Sublayer" (6top) is the next highest layer of the IEEE802.15.4e TSCH medium access control layer. It implements and terminates the "6top Protocol" (6P), and contains a "6top Scheduling Function" (SF). It is defined in TODO_LINK_draft-wang-6tisch-6top-sublayer.

SF: The "6top Scheduling Function" (SF) is the policy inside the "6TiSCH Operation Sublayer" (6top) which decides when to add/remove cells. It is defined in TODO_LINK_draft-wang-6tisch-6top-sublayer.

SFID: The "6top Scheduling Function Identifier" (SFID) is a 4-bit field identifying a SF. It is defined in TODO_LINK_draft-wang-6tisch-6top-sublayer.

6P: The "6top Protocol" (6P) allows neighbor nodes to communicate to add/delete cells to one another in their TSCH schedule. It is defined in TODO_LINK_draft-wang-6tisch-6top-sublayer.

6P Transaction: Part of the "6top Protocol" (6P), the action of two neighbors exchanging a 6P request message and the corresponding 6P response message. It is defined in TODO_LINK_draft-wang-6tisch-6top-sublayer.

Appendix D. [TEMPORARY] Changelog

o -03

  *
  * https://bitbucket.org/6tisch/draft-wang-6tisch-6top-sublayer/issues/32/missing-command-list
  * https://bitbucket.org/6tisch/draft-wang-6tisch-6top-sublayer/issues/31/missing-command-count
  * https://bitbucket.org/6tisch/draft-wang-6tisch-6top-sublayer/issues/30/missing-command-clear
  * https://bitbucket.org/6tisch/draft-wang-6tisch-6top-sublayer/issues/37/6top-atomic-transaction-6p-transaction

      * https://bitbucket.org/6tisch/draft-wang-6tisch-6top-sublayer/issues/35/separate-opcode-from-rc
      * https://bitbucket.org/6tisch/draft-wang-6tisch-6top-sublayer/issues/36/add-length-field-in-ie
      * https://bitbucket.org/6tisch/draft-wang-6tisch-6top-sublayer/issues/27/differentiate-rc_err_busy-and
      * https://bitbucket.org/6tisch/draft-wang-6tisch-6top-sublayer/issues/29/missing-rc-rc_reset
      * https://bitbucket.org/6tisch/draft-wang-6tisch-6top-sublayer/issues/28/the-sf-must-specify-the-behavior-of-a-mote
      * https://bitbucket.org/6tisch/draft-wang-6tisch-6top-sublayer/issues/26/remove-including-their-number
      * https://bitbucket.org/6tisch/draft-wang-6tisch-6top-sublayer/issues/34/6of-sf
      * https://bitbucket.org/6tisch/draft-wang-6tisch-6top-sublayer/issues/33/add-a-figure-showing-the-negociation

   o -02

      * introduces the 6P protocol and the notion of 6top Transaction.
      * introduces the concept of 6OF and its 6OFID.

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