

6TiSCH
Internet-Draft
Intended status: Informational
Expires: September 22, 2016

D. Dujovne, Ed.
Universidad Diego Portales
LA. Grieco
Politecnico di Bari
MR. Palattella
University of Luxembourg
N. Accettura
University of California Berkeley
March 21, 2016

6TiSCH 6top Scheduling Function Zero (SF0)
draft-dujovne-6tisch-6top-sf0-01

Abstract

This document defines a 6top Scheduling Function called "Scheduling Function Zero" (SF0). SF0 dynamically adapts the number of reserved cells between neighbor nodes, based on the currently allocated bandwidth and the neighbour nodes' requirements. Neighbor nodes negotiate in a distributed neighbor-to-neighbor basis the cell(s) to be added/deleted. SF0 uses the 6P signaling messages to add/delete cells in the schedule. Some basic rules for deciding when to add/delete cells and for selecting the cells to be added/deleted within the schedule are also provided.

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 September 22, 2016.

Copyright Notice

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

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

Table of Contents

1. Introduction	2
2. Scheduling Function Identifier	3
3. Rules for Adding/Deleting Cells	3
3.1. SF0 Triggering Events	3
3.2. SF0 Bandwidth Estimation Algorithm	3
3.3. SF0 Allocation Policy	4
4. Rules for CellList	5
5. 6P Timeout Value	6
6. Meaning of Metadata Information	6
7. Node Behavior at Boot	6
8. Relocating Cells	6
9. Forced Cell Deletion Policy	7
10. 6P Error Handling	7
11. Examples	7
12. Implementation Status	7
13. Security Considerations	8
14. IANA Considerations	8
15. Acknowledgments	8
16. References	8
16.1. Normative References	8
16.2. Informative References	9
Authors' Addresses	9

1. Introduction

This document defines the Scheduling Function for the 6top sublayer [I-D.wang-6tisch-6top-sublayer] called "Scheduling Function Zero" (SF0).

This document addresses the requirements for a scheduling function listed in [I-D.wang-6tisch-6top-sublayer], Section 4.2, and follows the recommended outline from Section 4.3.

2. Scheduling Function Identifier

The Scheduling Function Identifier (SFID) of SF0 is IANA_SFID_SF0.

3. Rules for Adding/Deleting Cells

A node running SF0 determines when to add/delete cells in a three-step process:

1. It waits for a triggering event (Section 3.1).
2. It applies the Bandwidth Estimation Algorithm (BEA) for a particular neighbor to determine how much bandwidth is required to that neighbor (Section 3.2).
3. It applies the Allocation Policy to compare the number of required cells to the number of already scheduled cells, and determine the number of cells to add/delete (Section 3.3).

3.1. SF0 Triggering Events

We RECOMMEND SF0 to be triggered at least by the following events:

1. If the Remaining Available Bandwidth (RAB) is less than the Minimum Remaining Bandwidth (MRB)
2. If there is any New Incoming Bandwidth Requirements from neighbour nodes (NIBR)

This allows SF0 to be triggered by any change in local node bandwidth and/or incoming bandwidth. The exact mechanism of when SF0 is triggered is implementation-specific.

3.2. SF0 Bandwidth Estimation Algorithm

The Bandwidth Estimation Algorithm takes into account the sum of the incoming bandwidth requirements from the neighbour nodes and the used outgoing bandwidth. This allows the node to estimate the total outgoing bandwidth requirement. As a consequence, the Bandwidth Estimation Algorithm for SF0 follows the steps described below:

1. Collect the New Incoming Bandwidth Requirements from neighbour nodes (NIBR)
2. Obtain the Current Outgoing Bandwidth Usage (COBU)
3. Obtain the number of Current Scheduled Bandwidth (CSB)
4. Calculate the New Outgoing Bandwidth (NOB) as: $NOB = COBU + NIBR$
5. Calculate the Remaining Available Bandwidth (RAB) as $RAB = CSB - NOB$

6. If the RAB is less than the Minimum Remaining Bandwidth (MRB),
Add MRB to the NOB: $NOB = NOB + MRB$
7. Submit the request to the allocation policy
8. Return to step 1 and wait for a triggering event.

3.3. SF0 Allocation Policy

The "Allocation Policy" is the set of rules used by SF0 to decide when to add/delete cells to a particular neighbor to satisfy the bandwidth requirements.

SF0 uses the following parameters:

SCHEDULEDCELLS: The number of cells scheduled from the current node to a particular neighbor.

REQUIREDCELLS: The number of cells calculated by the Bandwidth Estimation Algorithm from the current node to that neighbor.

SF0THRESH: Threshold parameter introducing cell over-provisioning in the allocation policy. It is a non-negative value expressed as number of cells. The definition of this value is implementation-specific; however, it is RECOMMENDED a SF0THRESH value of 3 cells. A setting of $SF0THRESH > 0$ will cause the node to allocate at least SF0THRESH cells to each of its' neighbours.

The SF0 allocation policy compares REQUIREDCELLS with SCHEDULEDCELLS and decides to add/delete cells taking into account SF0THRESH. This is illustrated in Figure 1.

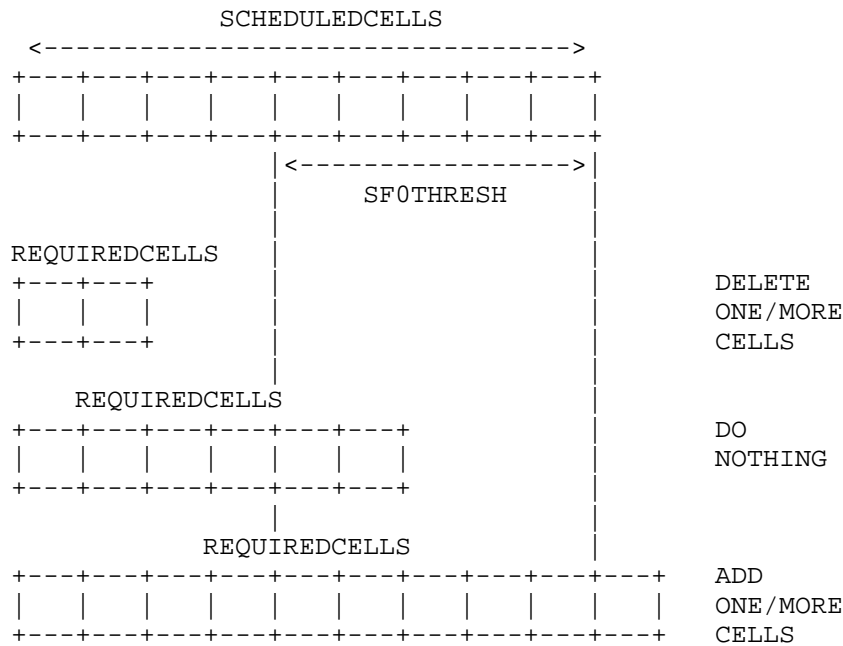


Figure 1: The SF0 Allocation Policy

1. If $\text{REQUIREDCELLS} < (\text{SCHEDULEDCELLS} - \text{SF0THRESH})$, delete one or more cells.
2. If $(\text{SCHEDULEDCELLS} - \text{SF0THRESH}) \leq \text{REQUIREDCELLS} \leq \text{SCHEDULEDCELLS}$, do nothing.
3. If $\text{SCHEDULEDCELLS} \leq \text{REQUIREDCELLS}$, add one or more cells.

When SF0THRESH equals 0, any discrepancy between REQUIREDCELLS and SCHEDULEDCELLS triggers an action to add/delete cells. Positive values of SF0THRESH reduce the number of 6P Transactions.

The Allocation Policy also translates the bandwidth requirement into cells according to their PDR. For example, if a cell with a 100% PDR is equivalent to 1Kbps, and the required bandwidth is 8Kbps, then, the number of scheduled cells will be 8. However, if two of the allocated cells have a 70% PDR, there number of scheduled cells will be 9.

4. Rules for CellList

When issuing a 6top ADD Request, SF0 executes the following sequence:

Whitelist case:

The Transaction Source node: Prepares the CellList field by selecting randomly the required cells, verifying that the slot offset and channel offset are not occupied.

The Transaction Destination node: Goes through the cells in the CellList in order, verifying whether there are no slotOffset conflicts.

Blacklist case:

The Transaction Source node: Prepares the CellList field by building a list of currently scheduled cells into the CellList.
The Transaction Destination node: Selects randomly the required cells, verifying that the slot offset and channel offset are not occupied from the ones on the CellList.

5. 6P Timeout Value

The 6P Timeout Value provided by SF0 allows the maximum number of TSCH link-layer retries. Given the TSCH parameters for the backoff mechanism, `macMinBE` and `macMaxBE`, and the length in seconds of the minimal Slotframe, `SM`, the timeout value is computed as: $\text{timeout} = (2^{(\text{macMaxBE}+1)} - 2^{\text{macMinBE}}) * \text{SM}$ TODO: Change general timeout to a timeout adapted to the schedule: SF to use the number of slots until the next scheduled cell.

6. Meaning of Metadata Information

The Metadata 16-bit field is used as follows:

BITS 0-7 [SLOTFRAME] are used to identify the slotframe number
BITS 8-14 are RESERVED
BIT 15 [WBLIST] is used to indicate that the CellList provided is a Whitelist (value=0) or a Blacklist (value=1).

TODO: length of the SlotFrame SHOULD be an integer multiple of the length of the minimal SlotFrame.

7. Node Behavior at Boot

In order to define a known state after the node is restarted, a CLEAR command is issued to each of the neighbour nodes to enable a new allocation process. TODO: Temporary cells from a pool for the join process.

8. Relocating Cells

SF0 uses Packet Delivery Rate (PDR) statistics to monitor the currently allocated cells for cell re-allocation (by changing their

slotOffset and/or channelOffset) when it finds out that the PDR of one or more softcells below 20% of the average PDR.

9. Forced Cell Deletion Policy

TODO: When all the cells are scheduled, we need a policy to free cells, for example, under alarm conditions or if a node disappears from the neighbour list.

10. 6P Error Handling

A node implementing SF0 handles a 6P Response depending on the Return Code it contains:

RC_SUCCESS:

If the number of elements in the CellList is the number of cells specified in the NumCells field of the 6P ALL Request, the operation is complete. The node does not take further action. If the number of elements in the CellList is smaller (possibly 0) than the number of cells specified in the NumCells field of the 6P ALL Request, the neighbor has received the request, but less than NumCells of the cells in the CellList were. In that case, the node MAY retry immediately with a different CellList if the amount of storage space permits, or build a new (random) CellList.

RC_ERR_VER: The node MUST NOT retry immediately. The node MAY add the neighbor node on a blacklist. The node MAY retry to contact this neighbor later.

RC_ERR_6OFID: The node MUST NOT retry immediately. The node MAY add the neighbor node on a blacklist. The node MAY retry to contact this neighbor later.

RC_ERR_NORESOURCES: Wait for a timeout and restart the scheduling process.

RC_ERR_BUSY: Issue a RESET command.

11. Examples

TODO

12. 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.

13. Security Considerations

TODO

14. IANA Considerations

- o IANA_SFID_SF0

15. Acknowledgments

Thanks to Kris Pister for his contribution in designing the default Bandwidth Estimation Algorithm. Thanks to Qin Wang and Thomas Watteyne for their support in defining the interaction between SF0 and the 6top sublayer.

This work is partially supported by the Fondecyt 1121475 Project, the Inria-Chile "Network Design" group, and the IoT6 European Project (STREP) of the 7th Framework Program (Grant 288445).

16. References

16.1. 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.

[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.

16.2. 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-terminology]

Palattella, M., Thubert, P., Watteyne, T., and Q. Wang, "Terminology in IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-terminology-07 (work in progress), March 2016.

[I-D.wang-6tisch-6top-sublayer]

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

Authors' Addresses

Diego Dujovne (editor)
Universidad Diego Portales
Escuela de Informatica y Telecomunicaciones
Av. Ejercito 441
Santiago, Region Metropolitana
Chile

Phone: +56 (2) 676-8121
Email: diego.dujovne@mail.udp.cl

Luigi Alfredo Grieco
Politecnico di Bari
Department of Electrical and Information Engineering
Via Orabona 4
Bari 70125
Italy

Phone: 00390805963911
Email: a.grieco@poliba.it

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

Nicola Accettura
University of California Berkeley
Berkeley Sensor & Actuator Center
490 Cory Hall
Berkeley, California 94720
USA

Email: nicola.accettura@eecs.berkeley.edu

6tisch
Internet-Draft
Intended status: Standards Track
Expires: September 19, 2016

S. Anamalamudi
M. Zhang
Huawei Technologies
C. Perkins
Futurewei
D. Liu
China Telcom Co., Ltd
S.V.R.Anand
Indian Institute of Science
March 18, 2016

Asymmetrical AODV-P2P-RPL in 6tisch Networks
draft-satish-6tisch-aodv-rpl-00

Abstract

Asymmetrical link based time-sensitive traffic flows with highly reliable shortest end-to-end route discovery is pre-requisite in IPv6 over the TSCH mode of IEEE 802.15.4e (6tisch) Networks. To achieve, this document specifies a resource reservation based reactive P2P route discovery mechanism for hop-by-hop routing (storing mode) based on Ad Hoc On-demand Distance Vector Routing (AODV) RPL protocol for 6tisch Networks. Two separate instances are used to achieve directional paths based on asymmetric links in between source and destination.

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 September 19, 2016.

Copyright Notice

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

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

Table of Contents

1. Introduction	2
2. Terminology	4
3. Overview of AODV-RPL	5
4. AODV Route Discovery Mode	5
4.1. Route Discovery mode for DIO-RREQ-Instance-1	8
4.2. Route Discovery mode for DIO-RREQ-Instance-2	10
5. Resource reservation for P2P Communication at 6TOP	12
5.1. Operation of Trickle Timer	14
6. IANA Considerations	14
6.1. Additions to Mode of Operation	14
6.2. Additions to RPL Control Message Options	14
7. Security Considerations	15
8. References	15
8.1. References	15
8.2. Informative References	16
Authors' Addresses	17

1. Introduction

Deterministic Networks enable time sensitive traffic flows that are highly sensitive to jitter, quite sensitive to latency, and with a high degree of operational criticality. This clearly depicts that nodes in the Deterministic networks need to be scheduled to support critical packet flows. To achieve this, 6TiSCH Working Group focus on the Time Slotted Channel Hopping (TSCH) mode of the IEEE802.15.4e standard to schedule traffic flows through Channel Distribution and Usage (CDU) matrix. [I-D.ietf-6tisch-minimal] describes about initial formation of 6tisch network during network bootstrap through Enhanced Beacons (EB). RPL [RFC6550], the IPv6 distance vector routing protocol for Low-power and Lossy Networks (LLNs), is used on the resulting 6tisch network. RPL is designed to support multiple

traffic flows through a root-based Destination Oriented Directed Acyclic Graph (DODAG). For Point-to-Point (P2P) traffic flows (meaning that two routers within the RPL network need to communicate), this means that data packets either have to traverse the root in non-storing mode (source routing), or traverse a common ancestor in storing mode (hop-by-hop routing). Such P2P traffic thereby flows along sub-optimal routes between arbitrary router pairs and may suffer severe traffic congestion near the DAG root [RFC6997], [RFC6998]. This sub-optimal paths in RPL [RFC6550] result in increased resource reservation control overhead (6top control message overhead) and in-efficient bandwidth allocation (cells) for P2P traffic flows in 6tisch networks. To avoid this issue, it is desirable for child node to acquire resources (cells) reactively from its next hop neighbor (temporary parent) towards destination instead of original parent of RPL. In addition, severe traffic congestion near the DAG root MAY leads to increased packet drops that need to be taken care more efficiently for time-sensitive traffic flows in 6tisch networks.

To overcome sub-optimal paths for P2P traffic flows in RPL, P2P-RPL [RFC6997] is proposed with a temporary DODAG where the source acts as temporary root. The source initiates "P2P Route Discovery mode (P2P-RDO)" with "address vector" for both non-storing mode (H=0) and storing mode (H=1). Subsequently, each intermediate router will add its IP address and multicast the P2P-RDO message again, until it reaches the destination. The destination will send "Discovery reply option", using either "hop-by-hop routing" or "source routing", based on "H" field in P2P-RDO. The proposed solution is efficient for source routing, but much less efficient for hop-by-hop routing. This is due to the extra address vector overhead in hop-by-hop routing. In fact, when the P2P-RDO message is being multicast from the source hop-by-hop, receiving nodes are able to figure out a next hop towards the source in symmetric links. Subsequently, when the destination replies to the source along the established routes, receiving nodes can once again figure out the next hop towards the destination. In other words, it is efficient to use only routing tables for P2P-RDO message instead of "Address vector" for purely hop-by-hop routes (H=1) in symmetrical links.

Both RPL and P2P-RPL is proposed for single DODAG where bi-directional symmetrical links are assumed. But, application-specific routing requirements that are defined in IETF ROLL Working Group [RFC5548], [RFC5673], [RFC5826] and [RFC5867] may have routing metrics and routing constraints that refer to links with bi-directional asymmetric properties. To achieve this, [I-D.thubert-roll-asymlink] describes about bi-directional asymmetrical links for RPL [RFC6550] with Paired DODAGs where the DAG root (DODAGID) is common for two Instances. This satisfies the

application-specific routing requirements for bi-directional asymmetrical links in root based RPL [RFC6550]. However, P2P-RPL [RFC6997] for Paired DODAGs may need to have two DAG roots: one for the source and the other for the destination due to on-demand temporary DODAG formation. Moreover, applications in deterministic networks [I-D.grossman-detnet-use-cases] MAY also need to allocate asymmetrical links for P2P traffic flows where resource reservation (cell allocation) is different for bi-directional links. To achieve, this document specifies P2P route discovery through AODV-RPL, given the network supports bi-directional asymmetric links (See Section 4) and describes how 6top reserves resources (See Section 5) required by the discovered route [I-D.wang-6tisch-6top-sublayer]. This scenario requires two multicast messages to discover routes for bi-directional asymmetric links. With AODV-RPL, there is no "Address vector" control overhead during route discovery for paired DODAG scenarios. It is noteworthy that proposed AODV-RPL is designed on the top of the RPL routing protocol[RFC6550].

The main objective of AODV-RPL with bi-directional asymmetric links is to discover P2P routes reactively rather than those available along a global DAG [I-D.thubert-roll-asymlink]. The discovered routes of each bi-directional path must meet the application specific metrics and constraints that are separately defined in each Objective Function for each instance [RFC6552]. In this specification, all the nodes within the constrained network are required to support both instances to enable on-demand route establishment.

2. Terminology

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 [RFC2119]. Additionally, this document uses the following terms:

AODV

Ad Hoc On-demand Distance Vector Routing[RFC3561].

Source

The IPv6 router initiating the AODV-RPL route discovery.

Destination

The IPv6 router at the other end point of the P2P route(s) within the LLN network.

Bi-directional Asymmetric Link

A link that can be used in both directions but with different link characteristics. [I-D.thubert-roll-asymlink]

Instance-1

Instance used for control transmission from Source to Destination and data transmission from Destination to Source.

Instance-2

Instance used for control transmission from Destination to Source and data transmission from Source to Destination.

hop-by-hop routing

Routing when each node stores routing information about the next hop towards Source or Destination.

Paired DODAGs

Two DODAGs for a single application.

P2P

Point-to-Point.

source routing

The mechanism by which the source supplies the complete route to the destination along with each data packet [RFC6997].

3. Overview of AODV-RPL

With AODV-RPL, routes from Source to Destination within the LLN network are "on-demand"; in other words, the route discovery mechanism in AODV-RPL will be performed reactively when source has data for delivery to the Destination but existing routes do not satisfy the application's requirements. Unlike base RPL [RFC6550] and P2P-RPL [RFC6997], AODV-RPL can determine routes in networks with bi-directional asymmetric links. In other words, AODV-RPL is designed to discover two routes namely one from Source to Destination and other from Destination to Source. In addition, AODV-RPL can also support purely symmetric links for Paired DODAGs through "A" bit that is explained in Section 4.

4. AODV Route Discovery Mode

In AODV-RPL, route discovery is initiated by forming a temporary DAG rooted at the Source. Paired DODAGs are used to achieve bi-directional asymmetrical link formation in between Source and Destination. AODV-RPL is designed to support two instances. Instance-1 is used for the route control messages from Source to Destination whereas Instance-2 is used for route control messages from Destination to Source (as shown in Figure 2). Intermediate routers join the Paired DODAGs based on the rank determined from the DIO message. Henceforth in this document, the DIO-RREQ-Instance-1 message represents the Route Discovery message from Source to

Destination whereas DIO-RREQ-Instance-2 represents the Route Discovery message from Destination to Source. Subsequently, Instance-1 is used for data transmission from Destination to Source and Instance-2 is used for Data transmission from Source to Destination. The operation of the discovery mechanism resembles base RPL, extended by a new option called AODV-RREQ in a modified DIO message [I-D.thubert-roll-asymlink]. A new bit called Asymmetric bit ("A") is added to the base DIO message as shown in Figure 1. Source will always set the "D" bit to 1 and "A" bit to 0 while initiating the DIO-RREQ-Instance-1 message.

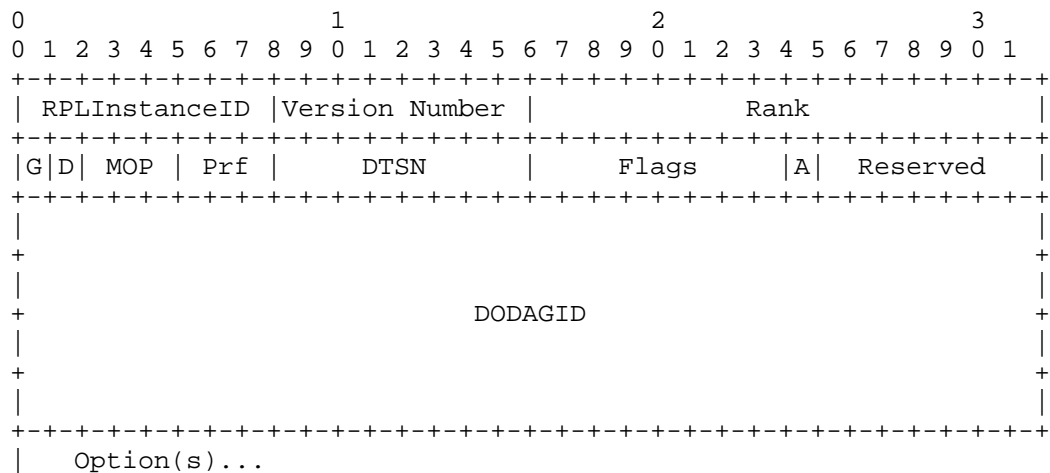
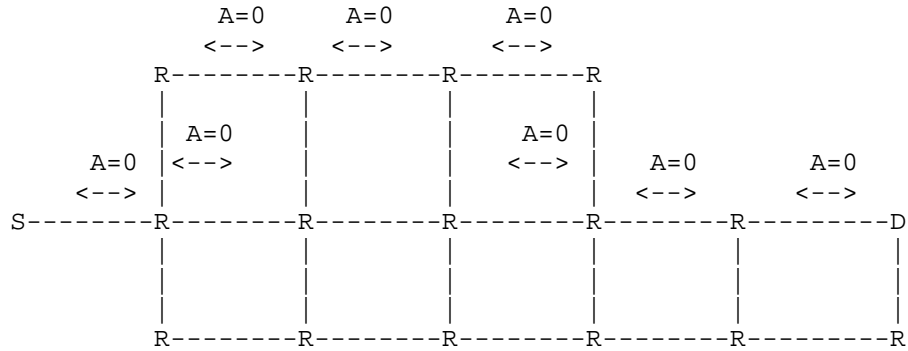


Figure 1: Modified DIO to support asymmetric links

The "D" bit in the DIO message indicates the directional DODAG information whereas "A" bit describes the link nature (Asymmetric or Symmetric). Figure 2 describes about operation of "A" bit for symmetrical and asymmetrical links. If the DIO-RREQ-Instance-1 arrives over an interface (Intermediate router) that is known to be symmetric, and the 'A' bit is set to 0, then it remains set at 0 (see Figure 2(a)). If the DIO-RREQ-Instance-1 arrives over an interface that is not known to be symmetric, or is known to be asymmetric, then the 'A' bit is set to be 1. If the 'A' bit arrives already set to be '1', it is set to be '1' on retransmission (Figure 2(b)). The 'A' bit is set to mean that the route is asymmetric. If any Intermediate router along the way believes that the incoming link is asymmetric, then the "A" bit is set to be 1 and remains set to be 1 all the way to the destination. Otherwise if there are no asymmetric links the "A" bit remains set to zero. Based on the "A" bit received by Destination in DIO-RREQ-Instance-1, link nature (Asymmetric or Symmetric) is decided to transmit DIO-RREQ-Instance-2 message back to Source.

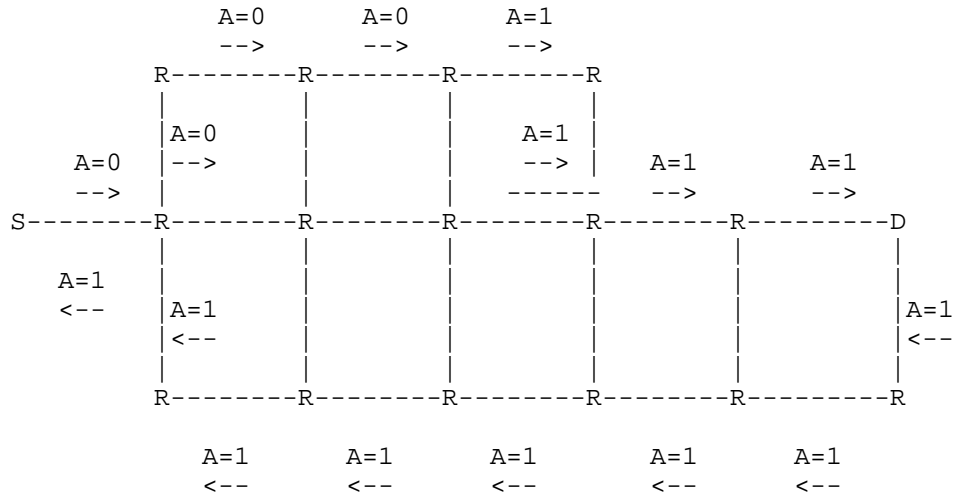
--Instance-1 (Control:S->D;Data:D->S) ----->



<---Instance-2(Control:D->S;Data:S->D)-----

(a). AODV-RPL with Symmetrical links

--Instance-1 (Control:S->D;Data:D->S) ----->



<---Instance-2(Control:D->S;Data:S->D)-----

(b). AODV-RPL with Asymmetrical links

S :Source
R :Intermediate nodes
D :Destination

Figure 2: AODV-RPL with Paired Instances

The value of 'A' bit (Symmetric or Asymmetric) can be decided by the available radio resources (cells) (See Section.5) during DIO-RREQ-Instance-1 message. Based on the received number of cell requests (NumCell) from ADDRequest in DIO-RREQ-Instance-1, Intermediate node will decide to set 'A' bit to '1' or remain to set 'A' bit to '0'. For example, if intermediate node has resource (cells) to transmit data for only one direction then it set A bit to '1'. If it has resources (cells) to support data in both directions then it can remain the 'A' bit to '0'. Even if there is atleast one link along the path of DIO-RREQ-Instance-1 that does not have cells for both directions then 'A' bit is set to '1' and Destination will start to multicast DIO-RREQ-Instance-2(see Figure 2(b)). If all the Intermediate nodes have cells for both directions then 'A' bit will be remain to '0' and DIO-RREQ-Instance-2 is unicast back in same path of DIO-RREQ-Instance-1 (see Figure 2(a)).

4.1. Route Discovery mode for DIO-RREQ-Instance-1

The AODV-RPL Source specifies the following information in the DIO-RREQ-Instance-1 message:

- D-bit

Directional bit in the DIO base object (D=1 for DIO-RREQ-Instance-1 message)[I-D.thubert-roll-asymlink].

- A-bit

Asymmetric bit in the DIO base object (A=0 for DIO-RREQ-Instance-1 message).

- MOP bit

MOP operation in the DIO object MUST be set to "5(tbd)" by Source for "AODV-RPL".

- RPLInstanceID

RPLInstanceID in the DIO object MUST be the InstanceID of Instance-1.

- DODAGID

DODAGID in the DIO object MUST be the IPv6 address of the Source that initiates the DIO-RREQ message of Instance-1.

- Rank

Rank in the DIO object MUST be the the rank of Instance-1.

- Metric Container Options

DIO-RREQ-Instance-1 message from Source MAY carry one or more Metric Container options to specify the relevant routing metrics.

- Destination address

IPv6 address of the Destination that receives DIO-RREQ-Instance-1 message. This address MUST be in the modified RREQ option (see Figure 3) of AODV [RFC3561].

- G bit

G(Gratuitous RREP flag) bit is set to "1" when Source has Destination Sequence number. When an intermediate node has a route towards destination with higher Destination Sequence number then Gratuitous DIO-RREP messages are unicast from the intermediate node to Destination. Note that Intermediate nodes never reply unicast Gratuitous DIO-RREP messages back to Source in Instance-1.

- J bit

Derived from [RFC3561]. Out of scope for this specification.

- R bit

Derived from [RFC3561]. Out of scope for this specification.

- D bit

Derived from [RFC3561]. Out of scope for this specification.

- U bit

Derived from [RFC3561]. Out of scope for this specification.

The Source in Figure 2 will multicast the DIO-RREQ-Instance-1 message (see Figure 3) to its one-hop neighbours. Intermediate nodes will compute the rank for Instance-1 and create a routing table entry for path towards the source if the routing metrics/constraints are satisfied. Subsequently, it checks for already existing path towards destination by comparing the Destination Sequence Numbers. Whenever the path exists from intermediate node to Destination, it unicast the Gratuitous DIO-RREP towards destination and creates the path towards Source for Instance-1. This helps to minimize the route control

message multicast overhead during Route Discovery process. The message format of Gratuitous DIO-RREP is same as [RFC3561] with the exception of the Source IP address which can be obtained through DODAGID of DIO base (see Figure 1). If the path towards Destination does not exist, the intermediate node has to re-multicast the DIO-RREQ-Instance-1 message with updated rank to the next-hop neighbours until the message reaches to Destination(Figure 2). Based on the "A" bit in received DIO-RREQ_instance-1, Destination will decide to unicast or multicast the DIO-RREQ-Instance-2 message back to Source.

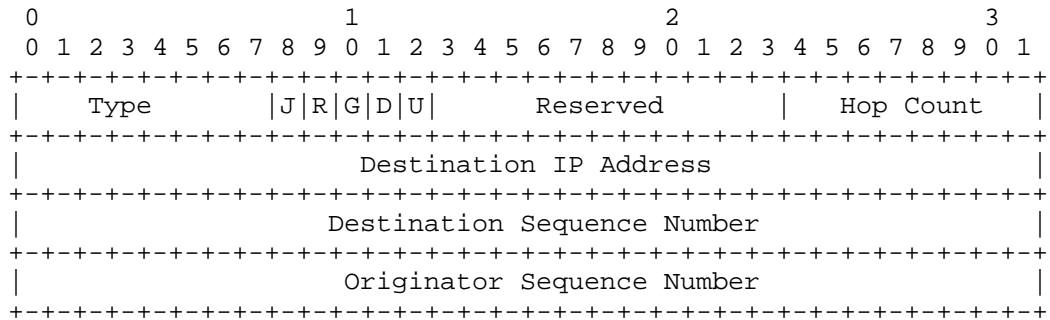


Figure 3: Modified DIO-RREQ message format

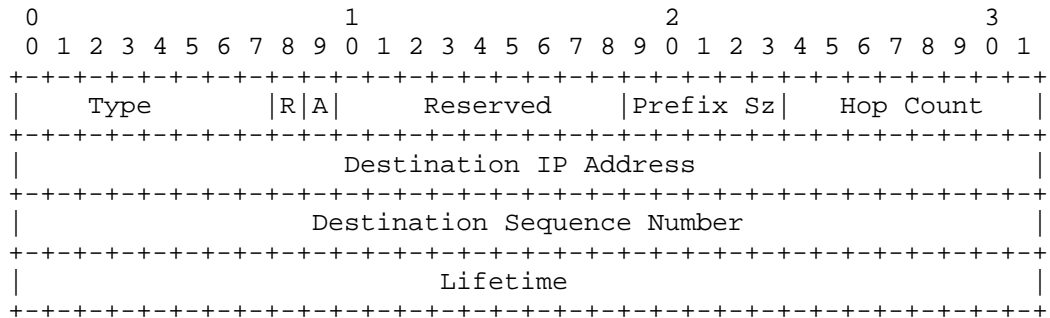


Figure 4: DIO-Gratuitous-RREP message format

4.2. Route Discovery mode for DIO-RREQ-Instance-2

The AODV-RPL Destination specifies the following information in the DIO-RREQ-Instance-2 message:

- D-bit

Directional bit in the DIO base object (D=1 for DIO-RREQ-Instance-2 message)[I-D.thubert-roll-asymlink].

- A-bit

Asymmetric bit in the DIO base object (value of "A" bit for DIO-RREQ-Instance-2 is directly copied from "A" bit of DIO-RREQ-Instance-1 message).

- MOP bit

MOP operation in the DIO object MUST be set to "5(tbd)" by Destination for "AODV-RPL".

- RPLInstanceID

RPLInstanceID in the DIO object MUST be the InstanceID of Instance-2.

- DODAGID

DODAGID in the DIO object MUST be the IPv6 address of the Destination that initiates the DIO-RREQ message of Instance-2.

- Rank

Rank in the DIO object MUST be the the rank of Instance-2.

- Metric Container Options

DIO-RREQ-Instance-2 message from Destination MAY carry one or more Metric Container options to specify the relevant routing metrics.

- Destination address

IPv6 address of the Source that receives DIO-RREQ-Instance-2 message. This address MUST be in the modified RREQ option (see Figure 3) of AODV [RFC3561].

- G bit

G(Gratuitous RREP flag) bit is set to "1" when Destination has Source Sequence number. When an Intermediate node has a route towards Source with higher Source Sequence number then Gratuitous DIO-RREP messages are unicast from Intermediate node to Source. Note that Intermediate nodes never reply unicast DIO-RREP messages back to Destination in Instance-2.

- J bit

Derived from [RFC3561]. Out of scope for this specification.

- R bit

Derived from [RFC3561]. Out of scope for this specification.

- D bit

Derived from [RFC3561]. Out of scope for this specification.

- U bit

Derived from [RFC3561]. Out of scope for this specification.

The Destination in Figure 2 start to multicast the DIO-RREQ-Instance-2 message when the received "A" bit in DIO-RREQ-Instance-1 is set to 1. Intermediate nodes will create the routing tables for the path towards the Destination during DIO-RREQ-Instance-2 messages to Source. If the intermediate nodes have path towards the Source, then it unicast the Gratuitous DIO-RREP towards the Source and creates the path towards Destination for Instance-2. Once the route control message is reached to Source, it will start transmitting the application data packets to the Destination in the path that is discovered through DIO-RREQ-Instance-2 messages. Similarly, application data from Destination to Source is transmitted through the path that is discovered from DIO-RREQ-Instance-1 message.

The Destination in Figure 2 start to unicast the DIO-RREQ-Instance-2 message when the received "A" bit in DIO-RREQ-Instance-1 is set to 0. In this case, route control messages and application data in between Source and Destination for both Instance-1 and Instance-2 are transmitted in symmetrical links.

5. Resource reservation for P2P Communication at 6TOP

Whenever, Source has data to destination it runs the Bandwidth Estimation Algorithm (BEA)[I-D.dujovne-6tisch-6top-sf0] to estimate the application bandwidth requirement and map it to required number of cells. Subsequently, 6P ADD Request [I-D.wang-6tisch-6top-sublayer] is appended to the DIO-RREQ-Instance-1 with NumCells equal to application bandwidth requirement that is known through BEA. It is noteworthy that CellList (slotoffset, channeloffset) and Container information in 6P ADD Request is set to zero during DIO-RREQ-Instance-1 multicast from Source. Once the DIO-RREQ-Instance-1 with 6P ADD Request is reached to intermediate node, it checks the NumCells field. When an Intermediate node is able to allocate its transmit and receive cells that are equal to NumCells of 6P ADD Request, then it is eligible to re-multicast the DIO-RREQ-Instance-1 with 6P ADD Request. At this point, link nature (Symmetrical or Asymmetrical) is decided by

Intermediate node based on the available resources (cells). If an Intermediate node has transmit and receive cells for both directions that are greater than or equal to NumCells of 6P ADD Request then 'A' bit is remain to set to '0'. If an intermediate node has cells available only for one direction (Destination to Source) then 'A' bit is set to '1' and it re-multicast the DIO-RREQ-Instance-1. Whenever, the intermediate node has Destination Sequence number that is greater than Sequence number specified in modified-RREQ then Gratuitous-RREP is unicast with 6P ADD Request. It is assumed that Intermediate nodes who know the path towards the Destination must be able to allocate both transmit and receive cells that are specified in NumCells to either single direction (Asymmetric) or both directions (Symmetric). Once the DIO-RREQ-Instance-1 with 6P ADD Request reaches to the Destination, it checks the 'A' bit to reply the DIO-RREQ_Instance-2 messages.

For Asymmetrical links(A=1) to Instance-1, it is notable that Source will allocate only receive cells, Destination will allocate only transmit cells and Intermediate nodes that multicast route control messages will allocate both transmit and receive cells to perform data transmission from Destination to Source in Instance-1.

For asymmetrical links (A=1) to Instance-2, Destination will estimate the application bandwidth requirement through BEA and map it to Numcell for DIO-RREQ-Instance-2. It is noteworthy that CellList(slotoffset, channeloffset) and Container information in 6P ADD Request is set to zero during DIO-RREQ-Instance-2 multicast from Destination. Intermediate nodes that are able to allocate it's both transmit and receive cells of Numcell in 6P ADD Request will re-multicast the DIO-RREQ-Instance-2 with 6P ADD Request until it reaches to Source. In this case, 'A' bit is always set to '1'. From this operation, it is notable that Source will allocate only transmit cells, Destination will allocate only receive cells and Intermediate nodes that multicast route control messages will allocate both transmit and receive cells to perform data transmission from Source to Destination in Instance-2.

Once the Source know the path towards the Destination in Instance-2, it performs the actual 6P negotiation (6P ADD Request, 6P ADD Response) [I-D.wang-6tisch-6top-sublayer] to request and allocates the CellList (slotoffset, channeloffset) hop-by-hop for actual data transmission. Similarly, Destination will perform 6P negotiation (6P ADD Request, 6P ADD Response) [I-D.wang-6tisch-6top-sublayer] to request and allocate the CellList(slotoffset, channeloffset) hop-by-hop towards Source in Instance-1 for data transmission. The cells(bundle) allocated for end-to-end path in Instance-1 is associated with one TrackID and the cells allocated for end-to-end path in Instance-2 is associated with another TrackID.

For asymmetrical links ($A=1$), allocation of transmit-receive cells for Instance-1 and allocation of transmit-receive cells for Instance-2 will be in different paths.

For symmetrical links ($A=0$), Source and Destination will use same path for both Instance-1 and Instance-2 to transmit data. Hence, allocation of transmit-receive cells for Instance-1 and allocation of transmit-receive cells for Instance-2 need to be in same path.

With AODV-RPL, the address vector is not required and resource reservation (cell allocation) is on-demand during reactive route-discovery. This efficiently utilizes the control packet size and radio resources that is most significant in 6tisch networks.

5.1. Operation of Trickle Timer

The operation of Trickle timer to control DIO-RREQ-Instance 1/ Instance-2 message is similar to P2P-RPL Trickle operation [RFC6997].

6. IANA Considerations

6.1. Additions to Mode of Operation

IANA is required to assign a new Mode of Operation, named "AODV-RPL" for Point-to-Point(P2P) hop-by-hop routing under the RPL registry. The value of tbd1 is assigned from the "Mode of Operation" space [RFC6550].

Value	Description	Reference
tbd1(5)	AODV-RPL	This document

Figure 5: Mode of Operation

6.2. Additions to RPL Control Message Options

IANA is required to assign two entries for a new RPL options: "DIO-RREQ-Instance-1" and "DIO-RREQ-Instance-2" values of tbd2 (0x0a) and tbd3(0x0b) from the "RPL Control Message Options" space [RFC6550].

Value	Meaning	Reference
tbd2(0x0a)	DIO-RREQ-Instance-1	This document
tbd3(0x0b)	DIO-RREQ-Instance-2	This document

Figure 6: RPL Control Message Options

7. Security Considerations

This document does not introduce additional security issues to [RFC6550]. For general RPL security considerations, see [RFC6550].

8. References

8.1. 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>>.
- [RFC3561] Perkins, C., Belding-Royer, E., and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", RFC 3561, DOI 10.17487/RFC3561, July 2003, <<http://www.rfc-editor.org/info/rfc3561>>.
- [RFC5548] Dohler, M., Ed., Watteyne, T., Ed., Winter, T., Ed., and D. Barthel, Ed., "Routing Requirements for Urban Low-Power and Lossy Networks", RFC 5548, DOI 10.17487/RFC5548, May 2009, <<http://www.rfc-editor.org/info/rfc5548>>.
- [RFC5673] Pister, K., Ed., Thubert, P., Ed., Dwars, S., and T. Phinney, "Industrial Routing Requirements in Low-Power and Lossy Networks", RFC 5673, DOI 10.17487/RFC5673, October 2009, <<http://www.rfc-editor.org/info/rfc5673>>.
- [RFC5826] Brandt, A., Buron, J., and G. Porcu, "Home Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5826, DOI 10.17487/RFC5826, April 2010, <<http://www.rfc-editor.org/info/rfc5826>>.
- [RFC5867] Martocci, J., Ed., De Mil, P., Riou, N., and W. Vermeulen, "Building Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5867, DOI 10.17487/RFC5867, June 2010, <<http://www.rfc-editor.org/info/rfc5867>>.

- [RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <<http://www.rfc-editor.org/info/rfc6550>>.
- [RFC6552] Thubert, P., Ed., "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6552, DOI 10.17487/RFC6552, March 2012, <<http://www.rfc-editor.org/info/rfc6552>>.
- [RFC6997] Goyal, M., Ed., Baccelli, E., Philipp, M., Brandt, A., and J. Martocci, "Reactive Discovery of Point-to-Point Routes in Low-Power and Lossy Networks", RFC 6997, DOI 10.17487/RFC6997, August 2013, <<http://www.rfc-editor.org/info/rfc6997>>.
- [RFC6998] Goyal, M., Ed., Baccelli, E., Brandt, A., and J. Martocci, "A Mechanism to Measure the Routing Metrics along a Point-to-Point Route in a Low-Power and Lossy Network", RFC 6998, DOI 10.17487/RFC6998, August 2013, <<http://www.rfc-editor.org/info/rfc6998>>.

8.2. Informative References

- [I-D.dujovne-6tisch-6top-sf0]
Dujovne, D., Grieco, L., Palattella, M., and N. Accettura, "6TiSCH 6top Scheduling Function Zero (SF0)", draft-dujovne-6tisch-6top-sf0-00 (work in progress), October 2015.
- [I-D.grossman-detnet-use-cases]
Grossman, E., Gunther, C., Thubert, P., Wetterwald, P., Raymond, J., Korhonen, J., Kaneko, Y., Das, S., and Y. Zha, "Deterministic Networking Use Cases", draft-grossman-detnet-use-cases-01 (work in progress), November 2015.
- [I-D.ietf-6tisch-minimal]
Vilajosana, X. and K. Pister, "Minimal 6TiSCH Configuration", draft-ietf-6tisch-minimal-15 (work in progress), February 2016.
- [I-D.thubert-roll-asymlink]
Thubert, P., "RPL adaptation for asymmetrical links", draft-thubert-roll-asymlink-02 (work in progress), December 2011.

[I-D.wang-6tisch-6top-sublayer]

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

Authors' Addresses

Satish Anamalamudi
Huawei Technologies
No. 156 Beiqing Rd. Haidian District
Beijing 100095
China

Email: satish.anamalamudi@huawei.com

Mingui Zhang
Huawei Technologies
No. 156 Beiqing Rd. Haidian District
Beijing 100095
China

Email: zhangmingui@huawei.com

Charles E. Perkins
Futurewei
2330 Central Expressway
Santa Clara 95050
Unites States

Email: charliep@computer.org

Dongxin Liu
China Telcom Co., Ltd
109 West Zhongshan Ave, Tianhe District
Guangzhou 510630
China

Email: liudx@gsta.com

S.V.R Anand
Indian Institute of Science
Bangalore
560012
India

Email: anand@ece.iisc.ernet.in

6TiSCH
Internet-Draft
Intended status: Informational
Expires: September 21, 2016

Q. Wang, Ed.
Univ. of Sci. and Tech. Beijing
X. Vilajosana
Universitat Oberta de Catalunya
March 20, 2016

6top Protocol (6P)
draft-wang-6tisch-6top-protocol-00

Abstract

This document defines the 6top Protocol (6P), which enables distributed scheduling in 6TiSCH networks. 6P allows neighbor nodes in a 6TiSCH network to add/delete TSCH cells to one another. 6P is part of the 6TiSCH Operation Sublayer (6top), the next higher layer of the IEEE802.15.4 TSCH medium access control layer. The 6top Scheduling Function (SF) decides when to add/delete cells, and triggers 6P transactions. Several SFs can be defined, each identified by a different 6top Scheduling Function Identifier (SFID). This document lists the requirements for an SF, but leaves the definition of the SF out of scope. Different SFs are expected to be defined in future companion specifications.

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 September 21, 2016.

Copyright Notice

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

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

Table of Contents

1. TEMPORARY EDITORIAL NOTES	3
2. Introduction	3
3. 6TiSCH Operation Sublayer (6top)	5
3.1. Hard/Soft Cells	5
3.2. Using 6top with the Minimal 6TiSCH Configuration	5
4. 6top Protocol (6P)	6
4.1. 6top Transaction	6
4.1.1. 2-step 6top Transaction	6
4.1.2. 3-step 6top Transaction	7
4.2. Message Format	8
4.2.1. 6top Information Element	8
4.2.2. General Message Format	9
4.2.3. 6P Command Identifiers	9
4.2.4. 6P Return Codes	10
4.2.5. 6P Cell Format	10
4.2.6. 6P ADD Request Format	11
4.2.7. 6P DELETE Request Format	11
4.2.8. 6P COUNT Request Format	12
4.2.9. 6P LIST Request Format	12
4.2.10. 6P CLEAR Request Format	12
4.2.11. 6P Response Format	12
4.2.12. 6P Confirmation Format	13
4.3. Protocol Behavior	13
4.3.1. Version Checking	13
4.3.2. SFID Checking	13
4.3.3. Concurrent 6P Transactions	13
4.3.4. Timeout	14
4.3.5. SeqNum Mismatch	14
4.3.6. Adding cells	14
4.3.7. Aborting a 6P Transaction	15
4.3.8. Deleting cells	15

4.3.9. Handling error responses	15
4.4. Security	16
5. Guidelines for 6top Scheduling Functions (SF)	16
5.1. SF Identifier (SFID)	16
5.2. Requirements for an SF	16
5.3. Recommended Structure of an SF Specification	17
6. Implementation Status	17
7. Security Considerations	18
8. IANA Consideration	18
9. References	19
9.1. Normative References	19
9.2. Informative References	19
Appendix A. [TEMPORARY] IETF IE	20
Appendix B. [TEMPORARY] IEEE Liaison Considerations	20
Appendix C. [TEMPORARY] Terms for the Terminology Draft	20
Appendix D. [TEMPORARY] Changelog	21
Authors' Addresses	23

1. TEMPORARY EDITORIAL NOTES

This document is an Internet Draft, so work-in-progress by nature. It contains the following work-in-progress elements:

- o "TODO" statements are elements which have not yet been written by the authors for some reason (lack of time, ongoing discussions with no clear consensus yet, etc). The statement does indicate that the text will be written.
- o "TEMPORARY" Appendices are there to capture current ongoing discussions or the changelog of the document. These appendices will be removed in the final text.
- o "IANA_" identifiers are placeholders for numbers assigned by IANA. These placeholders are to be replaced by the actual values they represent after their assignment by IANA.
- o This section will be removed in the final text.

2. Introduction

All communication in a 6TiSCH network is orchestrated by a schedule [RFC7554]. This specification defines the 6top Protocol (6P), part of the 6TiSCH Operation Sublayer (6top) sublayer. 6P allow a node to communicate with a neighbor to add/remove a TSCH cell to one another. 6P hence enables distributed scheduling in a 6TiSCH network.

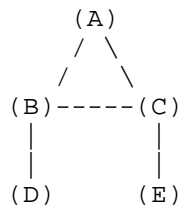


Figure 1: A simple 6TiSCH network.

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

- o 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 triggers a 6P transaction with node A to add one or more cells to A in the TSCH schedule.
- o If the traffic is lower than the capacity, node C triggers a 6P transaction with node A to delete one or more cells to A in the TSCH schedule.
- o 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 conceptually results in "relocating" the cell which suffered from collisions to a different slotOffset/channelOffset location in the TSCH schedule. The mechanism to handle cell relocation is out of the scope of this document and might be defined in a future document.

This results in distributed schedule management in a 6TiSCH network.

The 6top Scheduling Function (SF) defines when to add/delete a cell to a neighbor. The SF functions as a (required) add-on to 6P. Different applications require different SFs, so the SF is left out of scope of this document. Different SFs are expected to be defined in future companion specifications. A node MAY implement multiple SFs and run them at the same time. The SFID field contained in all 6P messages allows a node to switch between SFs on a per-transaction basis.

Section 3 describes the 6TiSCH Operation Sublayer (6top). Section 4 defines the 6top Protocol (6P). Section 5 provides guidelines on how to design an SF.

3. 6TiSCH Operation Sublayer (6top)

As depicted in Figure 2, the 6TiSCH Operation Sublayer (6top) is the next higher layer to the IEEE802.15.4 TSCH medium access control layer [IEEE802154-2015].

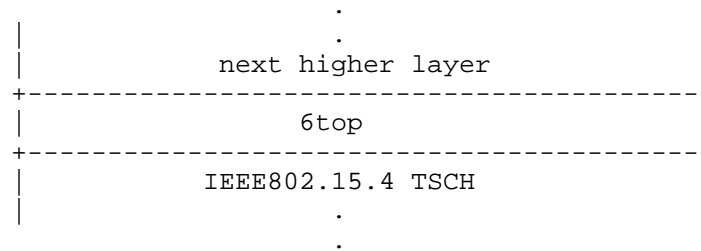


Figure 2: The 6top sublayer in the protocol stack.

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 one or more 6top Scheduling Function (SF), which define the algorithm to decide when to add/delete cells.

3.1. 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 scheduling. This is done, for example, in the Minimal 6TiSCH Configuration [I-D.ietf-6tisch-minimal].

3.2. 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 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 is used by 6top to allocate cells from. In Figure 3, this slotframe is 10 slots long, but it can be of any length.

Slotframe 0 SHOULD be of higher priority than Slotframe 1. 6top MAY support further slotframes; how to use more slotframes is out of the scope for this document.

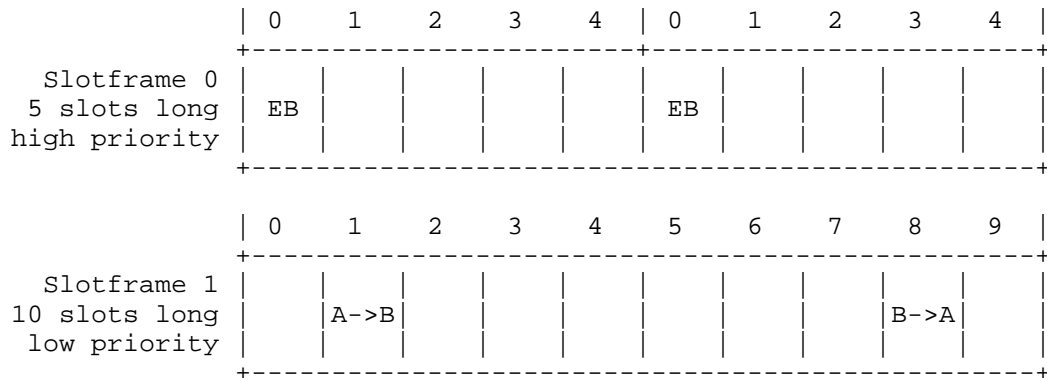


Figure 3: 2-slotframe structure when using 6top alongside the Minimal 6TiSCH Configuration.

4. 6top Protocol (6P)

The 6top Protocol (6P) allows two neighbor nodes to communicate to add/delete cells to their TSCH schedule. Conceptually, two neighbor nodes "negotiate" the location of the cell(s) to add/delete.

4.1. 6top Transaction

We call "6top Transaction" a complete negotiation between two neighbor nodes. A transaction starts when a node wishes to add/remove one or more cells to one of its neighbors; it ends when the cell(s) have been added removed from the schedule of both neighbor, or when the transaction has failed.

A transaction can consist of 2 or 3 steps. It is the SF which determines whether to use 2-step or 3-step transactions. An SF MAY use both 2-step and 3-step transactions.

We reuse the topology in Figure 1 to illustrate 2-step and 3-step transactions.

4.1.1. 2-step 6top Transaction

6P supports both 2- and 3-step transactions; the SF determinisms which to use. Without loss of generality, this section illustrates 2-step transaction through an example.

Figure 4 is a sequence diagram to help understand the core principle of 6P (several elements are left out to simplify understanding). We assume the SF running on node A determines 2 extra cells need to be scheduled to node B. In this example, node A proposes the cells to use.

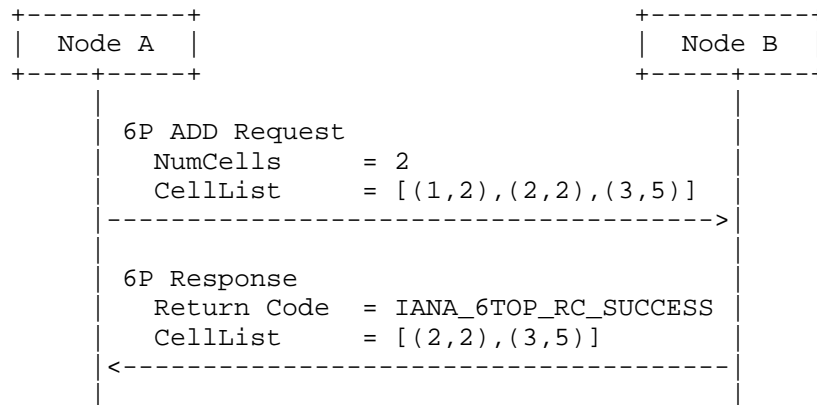


Figure 4: A 2-step 6P transaction.

In this example, the 2-step transaction occurs as follows:

1. The SF running on node A selects 3 candidate cells.
2. Node A sends a 6P ADD Request to node B, indicating it wishes to add 2 cells (the "NumCells" value), and specifying the list of 3 candidate (the "CellList" value). Each cell in the CellList is a (slotOffset, channelOffset) tuple.
3. The SF running on node B selects 2 of the 3 cells in the CellList of the 6P ADD Request. Node B sends back a 6P Response to node A, indicating the cells it selected.
4. The result of this 6P transaction is that 2 cells from A to B have been added to the TSCH schedule of both nodes A and B.

4.1.2. 3-step 6top Transaction

6P supports both 2- and 3-step transactions; the SF determinisms which to use. Without loss of generality, this section illustrates 3-step transaction through an example.

Figure 5 is a sequence diagram to help understand the core principle of 6P (several elements are left out to simplify understanding). We assume the SF running on node A determines 2 extra cells need to be scheduled to node B. In this example, node B proposes the cells to use.

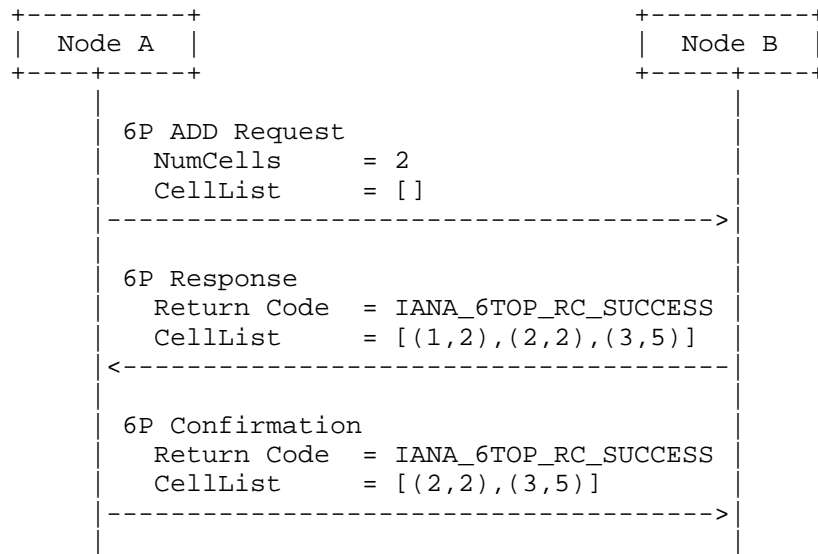


Figure 5: A 3-step 6P transaction.

In this example, the 3-step transaction occurs as follows:

1. The SF running on node A determines 2 extra cells need to be scheduled to node B, but does not select candidate cells.
2. Node A sends a 6P ADD Request to node B, indicating it wishes to add 2 cells (the "NumCells" value), with an empty "CellList".
3. The SF running on node B selects 3 candidate cells. Node B sends back a 6P Response to node A, indicating the 3 cells it selected.
4. The SF running on node B selects 2 cells. Node A sends back a 6P Confirmation to node B, indicating the cells it selected.
5. The result of this 6P transaction is that 2 cells from A to B have been added to the TSCH schedule of both nodes A and B.

4.2. Message Format

4.2.1. 6top Information Element

6P messages are carried as payload of IEEE802.15.4 Information Elements (IE) [IEEE802154-2015]. 6p messages travel over a single hop.

```

                                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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Payload IE Length | GroupID | T | Sub-ID | 6top IE Content
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Payload Termination IE |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

The 6top IE is an IETF IE with GroupID IANA_IETF_IE_GROUP_ID. The Sub-ID used by the 6top IE is IANA_6TOP_SUBIE_ID. The length of the 6top IE content is variable. The content of the 6top IE is specified in Section 4.2. The Payload Termination IE is defined by the IEEE802.15.4 standard [IEEE802154-2015]. TODO: IETF IE specified in Appendix A for now, but to be specified in a separate draft in the future.

4.2.2. General Message Format

In all 6P messages, the 6top IE content has 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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Version | Code | SFID | SeqNum | Other Fields...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Version (6P Version): The version of the 6P protocol. Only version IANA_6TOP_6P_VERSION is defined in this document. Future specifications 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_6TOP_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 5.1.

SeqNum: An identifier of the packet, used to match request and response. The value of SeqNum MUST increment by exactly one at each new 6P request issued to the same neighbor.

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

4.2.3. 6P Command Identifiers

Figure 6 lists the 6P command identifiers.

Value	Command ID	Description
IANA_6TOP_CMD_ADD	CMD_ADD	add one or more cells
IANA_6TOP_CMD_DELETE	CMD_DELETE	delete one or more cells
IANA_6TOP_CMD_COUNT	CMD_COUNT	count scheduled cells
IANA_6TOP_CMD_LIST	CMD_LIST	list the scheduled cells
IANA_6TOP_CMD_CLEAR	CMD_CLEAR	clear all cells
TODO-0xf	reserved	

Figure 6: 6P Command Identifiers

4.2.4. 6P Return Codes

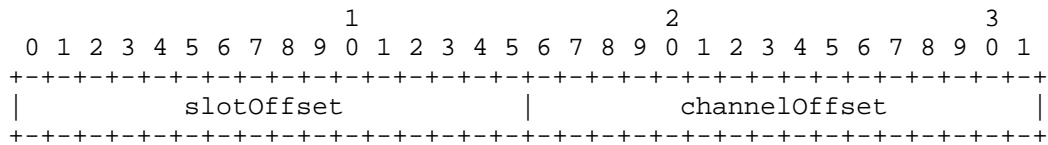
Figure 7 lists the 6P Return Codes and their meaning.

Value	Return Code	Description
IANA_6TOP_RC_SUCCESS	RC_SUCCESS	operation succeeded
IANA_6TOP_RC_VER_ERR	RC_VER_ERR	unsupported 6P version
IANA_6TOP_RC_SFID_ERR	RC_SFID_ERR	unsupported SFID
IANA_6TOP_RC_BUSY	RC_BUSY	handling previous request
IANA_6TOP_RC_RESET	RC_RESET	abort 6P transaction
IANA_6TOP_RC_ERR	RC_ERR	operation failed
TODO-0xf	reserved	

Figure 7: 6P Return Codes

4.2.5. 6P Cell Format

The 6P Cell is an element which is present in several messages. It is a 4-byte field, its RECOMMENDED format is:

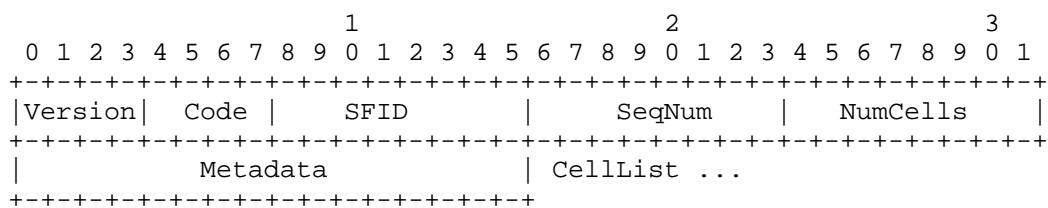


slotOffset: The slot offset of the cell.

channelOffset: The channel offset of the cell.

The CellList is an opaque set of bytes, sent unmodified to the SF. The SF MAY redefine the format of the CellList field.

4.2.6. 6P ADD Request Format



```
Version:  Set to IANA_6TOP_6P_VERSION.
```

Code: Set to IANA_6TOP_CMD_ADD for a 6P ADD Request.

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

SeqNum: Packet identifier to match 6P Request and 6P Response.

NumCells: The number of additional TX cells the sender wants to schedule to the receiver.

Metadata: Metadata used as extra signaling to the SF. One example use can be to specify which slotframe to schedule the cells to. The contents of the Metadata field is an opaque set of bytes, and passed unmodified 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 RECOMMENDED format of each 6P Cell is defined in Section 4.2.5. The CellList is an opaque set of bytes, sent unmodified to the SF. The SF MAY redefine the format of the CellList field.

4.2.7. 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 6TOP CMD DELETE.

4.2.8. 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 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|Version|  Code |    SFID      |    SeqNum    |   Metadata   |
+-----+-----+-----+-----+-----+-----+-----+-----+
|Metadata|      |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Version: Set to IANA_6TOP_6P_VERSION.

Code: Set to IANA_6TOP_CMD_COUNT for a 6P COUNT Request.

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

SeqNum: Packet identifier to match request and response.

Metadata: Metadata used as extra signaling to the SF. One example use can be to specify which slotframe to schedule the cells to. The contents of the Metadata field is an opaque set of bytes, and passed unmodified to the SF. The meaning of this field depends on the SF, and is hence out of scope of this document.

4.2.9. 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_6TOP_CMD_LIST.

4.2.10. 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_6TOP_CMD_CLEAR.

4.2.11. 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
+-----+-----+-----+-----+-----+-----+-----+-----+
|Version|  Code |    SFID      |    SeqNum    | Other Fields... |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Version: Set to IANA_6TOP_6P_VERSION.

SFID: Identifier of the SF to be used by the receiver to handle the message. The response MUST contain the same SFID value as the value in the SFID field of the 6P Request is responds to.

Code: One of the 6P Return Codes listed in Section 4.2.4.

SeqNum: Packet identifier to match request and response. The response MUST contain the same SeqNum value as the value in the SeqNum field of the 6P Request is responds to.

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 4.2.5.

Response to COUNT command: The number of cells scheduled from the requesting node to the receiver node by the 6P protocol, encoded as a 2-octet unsigned integer.

Response to CLEAR command: No other fields are present in the response.

4.2.12. 6P Confirmation Format

A 6P Confirmation is only used in a 3-step transaction, as the third step. A 6P Confirmation Message has the exact same format as a 6P Response Message. It is only the fact that it appears as the third step in a 3-step transaction that distinguishes it from a 6P Response. In particular, the same Return Codes are used in both 6P Response and 6P Confirmation messages.

4.3. 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.

4.3.1. 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_6TOP_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_6TOP_RC_VER_ERR. The Version field in the 6P Response MUST be the same as the Version field in the corresponding 6P Request.

4.3.2. 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_6TOP_RC_SFID_ERR. The Version field in the 6P Response MUST be the same as the Version field in the corresponding 6P Request.

4.3.3. 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_6TOP_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_6TOP_RC_BUSY.

4.3.4. 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.

4.3.5. SeqNum Mismatch

When a node receives a 6P Response with SeqNum value different from the SeqNum value in the 6P Request, it MUST drop the packet and consider the 6P Transaction as having failed.

4.3.6. 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_6TOP_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.

4.3.7. 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_6TOP_RC_RESET. Upon receiving this 6top reply, the initiator of the 6P Transaction MUST consider the 6P Transaction as failed.

4.3.8. 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.

4.3.9. Handling error responses

A return code with a name starting with "RC_ERR" in Figure 7 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.

4.4. 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.

5. Guidelines for 6top Scheduling Functions (SF)

5.1. 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
0x00-0xef	managed
0xf0-0xfe	unmanaged
0xff	reserved

Figure 8: SFID range.

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

5.2. Requirements for an SF

The specification for an SF

- o **MUST** specify an identifier for that SF.
- 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/equation to calculate it.
- o **MUST** specify a meaning for the "Metadata" 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 **MUST** specify the list of statistics to gather. An example statistic is the number of transmitted frames to each neighbor. In case the SF requires no statistics to be gathered, the specific of the SF **MUST** explicitly state so.

- o SHOULD clearly state the application domain the SF is created for.
- 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.
- o MAY redefine the format of the CellList field.

5.3. 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 the Metadata Field
- o Node Behavior at Boot
- o 6P Error Handling
- o Examples
- o Implementation Status
- o Security Considerations
- o IANA Considerations

6. 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".

ETSI 6TiSCH #2 plugtests: 6P was one of two protocols addressed during the ETSI 6TiSCH #2 plugtests organized on 2-4 February 2016 in Paris, France. 14 entities participated in this event, verifying the compliance and interoperability of their implementation of 6P. This event happened under NDA, so neither the name of the entities nor the test results are public. This event is, however, a clear indication of the maturity of 6P, and the interest it generates. More information about the event at <http://www.etsi.org/news-events/events/1022-6TiSCH-2-plugtests>.

OpenWSN: 6P is implemented in the OpenWSN project [OpenWSN] under a BSD open-source license. 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. More information about this implementation at <http://www.openwsn.org/>.

Wireshark Dissector: A Wireshark dissector for 6P is implemented under a BSD open-source license. It is not yet merged into the main Wireshark build, but can be downloaded at <https://github.com/openwsn-berkeley/dissectors/>.

7. Security Considerations

TODO: explicit 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.

8. IANA Consideration

TODO: write out this section as soon as the discussion with the IEEE about a possible IETF IE ID has concluded.

- o TODO: IANA_IETF_IE_GROUP_ID
- o TODO: IANA_6TOP_SUBIE_ID
- o TODO: IANA_6TOP_6P_VERSION
- o TODO: IANA_6TOP_CMD_ADD
- o TODO: IANA_6TOP_CMD_DELETE
- o TODO: IANA_6TOP_CMD_LIST
- o TODO: IANA_6TOP_CMD_COUNT
- o TODO: IANA_6TOP_CMD_CLEAR
- o TODO: IANA_6TOP_RC_SUCCESS
- o TODO: IANA_6TOP_RC_VER_ERR

- o TODO: IANA_6TOP_RC_SFID_ERR
- o TODO: IANA_6TOP_RC_BUSY
- o TODO: IANA_6TOP_RC_RESET
- o TODO: IANA_6TOP_RC_ERR

9. References

9.1. 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>>.

[IEEE802154-2015]
IEEE standard for Information Technology, "IEEE Std 802.15.4-2015 - IEEE Standard for Low-Rate Wireless Personal Area Networks (WPANs)", October 2015.

9.2. 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-15 (work in progress), February 2016.

[I-D.ietf-6tisch-terminology]
Palattella, M., Thubert, P., Watteyne, T., and Q. Wang, "Terminology in IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-terminology-06 (work in progress), November 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 than the 6P protocol defined in this document.

The proposal is to use an IETF IE, a IEEE802.15.4 Payload Information Element with the Group ID set to IANA_IETF_IE_GROUP_ID. The value of IANA_IETF_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 [IEEE802154-2015], Section 5.2.4.5. The difference is that the space of Sub-IDs is managed by the IETF/IANA. The Sub-ID used by 6top commands is IANA_6TOP_SUBIE_ID with value 0x00.

Other options are being discussed between the IETF 6TiSCH WG and the IEEE 6TiSCH IG, and listed in <https://www.ietf.org/mail-archive/web/6tisch/current/msg04469.html>. These options concern the way 6P Messages are transported as IEEE802.15.4 IEs, and do not impact the format of those messages.

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_IETF_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 IEEE802.15.4 TSCH medium access control layer. It implements and terminates the "6top Protocol" (6P), and contains one or more "6top Scheduling Function" (SF). It is defined in TODO_LINK_draft-wang-6tisch-6top-protocol.
- 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-protocol.

- SFID: The "6top Scheduling Function Identifier" (SFID) is a 1-byte field identifying a SF. It is defined in TODO_LINK_draft-wang-6tisch-6top-protocol.
- 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-protocol.
- 6P Transaction: Part of the "6top Protocol" (6P), we call "6top Transaction" a complete negotiation between two neighbor nodes. A transaction starts when a node wishes to add/remove one or more cells to one of its neighbors; it ends when the cell(s) have been added/removed from the schedule of both neighbor, or when the transaction has failed. It is defined in TODO_LINK_draft-wang-6tisch-6top-protocol.

Appendix D. [TEMPORARY] Changelog

- o draft-wang-6tisch-6top-protocol-00
 - * Editorial overhaul: fixing typos, increasing readability, clarifying figures.
 - * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/47>
 - * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/54>
 - * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/55>
 - * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/49>
 - * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/53>
 - * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/44>
 - * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/48>
 - * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/43>
 - * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/52>
 - * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/45>
 - * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/51>
 - * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/50>
 - * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/46>

- * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/41>
- * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/42>
- * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/39>
- * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/40>
- o draft-wang-6tisch-6top-sublayer-05
 - * Specifies format of IE
 - * Adds token in messages to match request and response
- o draft-wang-6tisch-6top-sublayer-04
 - * Renames IANA_6TOP_IE_GROUP_ID to IANA_IETF_IE_GROUP_ID.
 - * Renames IANA_CMD and IANA_RC to IANA_6TOP_CMD and IANA_6TOP_RC.
 - * Proposes IANA_6TOP_SUBIE_ID with value 0x00 for the 6top sub-IE.
- o draft-wang-6tisch-6top-sublayer-03
 - * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/32/missing-command-list>
 - * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/31/missing-command-count>
 - * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/30/missing-command-clear>
 - * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/37/6top-atomic-transaction-6p-transaction>
 - * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/35/separate-opcode-from-rc>
 - * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/36/add-length-field-in-ie>
 - * https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/27/differentiate-rc_err_busy-and
 - * https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/29/missing-rc-rc_reset
 - * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/28/the-sf-must-specify-the-behavior-of-a-mote>
 - * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/26/remove-including-their-number>
 - * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/34/6of-sf>
 - * <https://bitbucket.org/6tisch/draft-wang-6tisch-6top-protocol/issues/33/add-a-figure-showing-the-negotiation>
- o draft-wang-6tisch-6top-sublayer-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