

6TiSCH
Internet-Draft
Intended status: Experimental
Expires: March 26, 2018

D. Dujovne, Ed.
Universidad Diego Portales
LA. Grieco
Politecnico di Bari
MR. Palattella
Luxembourg Institute of Science and Technology (LIST)
N. Accettura
LAAS-CNRS
September 22, 2017

6TiSCH 6top Scheduling Function Zero / Experimental (SFX)
[draft-ietf-6tisch-6top-sfx-00](#)

Abstract

This document defines a Scheduling Function called "Scheduling Function Zero" (SF0) in its Experimental version. SF0 dynamically adapts the number of scheduled cells between neighbor nodes, based on the amount of currently allocated cells and the neighbor nodes' cell requirements. Neighbor nodes negotiate in a distributed neighbor-to-neighbor basis the number of cell(s) to be added/deleted. SF0 uses the 6P signaling messages to add/delete cells in the schedule. This function selects the candidate cells from the schedule, defines which cells will be added/deleted and triggers the allocation/deallocation process.

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

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

This Internet-Draft will expire on March 26, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](https://trustee.ietf.org/license-info) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://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.	Scheduling Function Identifier	4
4.	Allocated and Used Cells	4
5.	Overprovisioning	4
6.	Scheduling Algorithm	4
6.1.	SF0 Triggering Events	4
6.2.	SF0 Cell Estimation Algorithm	5
6.3.	SF0 Allocation Policy	6
7.	Rules for CellList	8
8.	6P Timeout Value	8
9.	Meaning of Metadata Information	9
10.	Node Behavior at Boot	9
11.	Cell Type	9
12.	SF0 Statistics	9
13.	Relocating Cells	9
14.	Forced Cell Deletion Policy	10
15.	6P Error Handling	10
16.	Experimental requirements	10
17.	Examples	11
18.	Implementation Status	11
19.	Security Considerations	11
20.	IANA Considerations	12
21.	6P Compliance	12
22.	Acknowledgments	12
23.	References	13
23.1.	Normative References	13
23.2.	Informative References	13
Appendix A.	[TEMPORARY] Changelog	13

Authors' Addresses	14
------------------------------	--------------------

1. TEMPORARY EDITORIAL NOTES

This document is an Internet Draft, so it is 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, etc). The statement does indicate that the text will be written at some time.
- 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 The string "REMARK" is put before a remark (questions, suggestion, etc) from an author, editor or contributor. These are on-going discussions at the time of writing, NOT part of the final text.
- o This section will be removed in the final text.

2. Introduction

This document defines a minimal Scheduling Function using the 6P protocol [[I-D.ietf-6tisch-6top-protocol](#)], called "Scheduling Function Zero" (SF0). SF0 is designed to offer a number of functionalities to be usable in a wide range of applications. SF0 defines two algorithms: The Scheduling Algorithm defines the number of cells to allocate/delete between two neighbours and the Relocation Algorithm defines when to relocate a cell.

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

1. It waits for a triggering event ([Section 6.1](#)).
2. It applies the Cell Estimation Algorithm (CEA) for a particular neighbor to determine how many cells are required to that neighbor ([Section 6.2](#)).
3. It applies the Allocation Policy to compare the number of required cells to the number of already scheduled cells, and determines the number of cells to add/delete ([Section 6.3](#)).

We expect additional SFs, offering more functionalities for a more specific use case, to be defined in future documents. SF0 addresses the requirements for a scheduling function listed in [Section 5.2](#) from [[I-D.ietf-6tisch-6top-protocol](#)], and follows the recommended outline listed in Section 5.3 of [[I-D.ietf-6tisch-6top-protocol](#)]. This

document follows the terminology defined in [\[I-D.ietf-6tisch-terminology\]](#).

3. Scheduling Function Identifier

The Scheduling Function Identifier (SFID) of SF0 is IANA_6TISCH_SFID_SF0.

4. Allocated and Used Cells

An allocated cell is assigned as a TX, RX or Shared cell on the schedule, as a reserved resource. This reservation does not imply that a packet will be transmitted during the scheduled cell time. A used cell is a cell where a packet has been transmitted during the scheduled cell time on the last slotframe.

5. Overprovisioning

Overprovisioning is the action and effect of increasing a value representing an amount of resources. In the case of SF0, overprovisioning is done as a provision to reduce traffic variability effects on packet loss, to the expense of artificially allocating a number of cells.

6. Scheduling Algorithm

A number of TX cells must be allocated between neighbor nodes in order to enable data transmission among them. A portion of these allocated cells will be used by neighbors, while the remaining cells can be over-provisioned to handle unanticipated increases in cell requirements. The Scheduling Algorithm collects the cell allocation/deallocation requests from the neighbors and the number of cells which are currently under usage. First, the Cell Estimation Algorithm calculates the number of required cells and second, the calculated number is transferred to the Allocation Policy. In order to reduce consumption, this algorithm is triggered only when there is a change on the number of used cells from a particular node.

6.1. SF0 Triggering Events

We RECOMMEND SF0 to be triggered at by the following event: If there is a change on the number of used cells towards any of the neighbours. The exact mechanism of when SF0 is triggered is implementation-specific.

6.2. SF0 Cell Estimation Algorithm

The Cell Estimation Algorithm takes into account the number of current used cells to the neighbour. This allows the algorithm to estimate a new number of cells to be scheduled to the neighbour. As a consequence, the Cell Estimation Algorithm for SF0 follows the steps described below:

1. Collect the current number of used cells to the neighbour
2. Calculate the new number of cells to be scheduled to the neighbour by adding the current number of used cells plus an OVERPROVISION number of cells
3. Transfer the request to the allocation policy as REQUIREDCELLS
4. Return to step 1 and wait for a triggering event.

The Cell Estimation Algorithm is depicted on figure Figure 1. The OVERPROVISION parameter is calculated as a percentage of the number of currently scheduled cells to the neighbour. OVERPROVISION is added to the amount of used cells to the neighbour to reduce the probability of packet loss given a sudden growth on the number of used cells to the neighbour. The OVERPROVISION value is implementation-specific. A value of OVERPROVISION equal to zero leads to queue growth and possible packet loss: In this case, there are no overprovisioned cells where a sudden growth on the number of cells can be absorbed and detected.

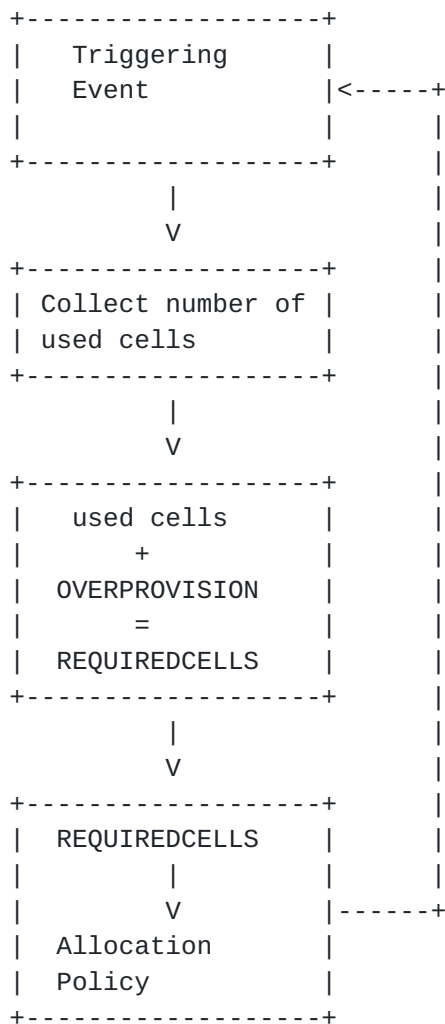


Figure 1: The SF0 Estimation Algorithm

6.3. 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 cell 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 Cell 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. A setting of $SF0THRESH > 0$ will cause the node to allocate at least $SF0THRESH$ cells to each of its' neighbors.

The $SF0$ allocation policy compares $REQUIREDCELLS$ with $SCHEDULEDCELLS$ and decides to add/delete cells taking into account $SF0THRESH$. This is illustrated in Figure 2. The number of cells to be added/deleted is out of the scope of this document and it is implementation-dependent.

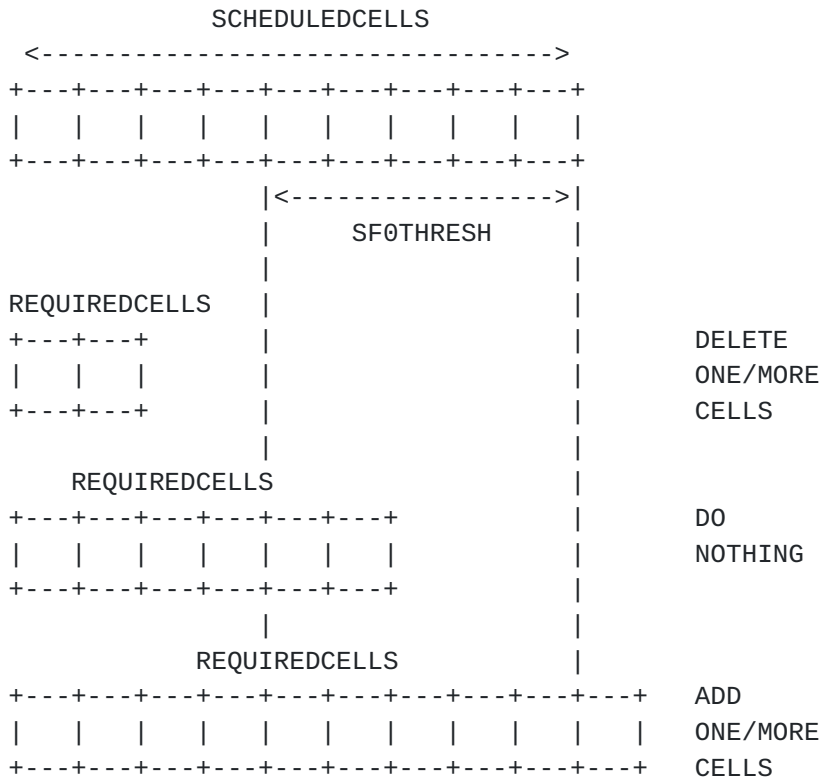


Figure 2: The $SF0$ Allocation Policy

1. If $REQUIREDCELLS < (SCHEDULEDCELLS - SF0THRESH)$, delete one or more cells.
2. If $(SCHEDULEDCELLS - SF0THRESH) \leq REQUIREDCELLS \leq SCHEDULEDCELLS$, do nothing.
3. If $SCHEDULEDCELLS < REQUIREDCELLS$, 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 number of cells to add or delete is implementation-specific.

7. Rules for CellList

There are two methods to define the CellList: The Whitelist method, which fills the CellList with the number of proposed cells to the neighbour, and the Blacklist, which fills the CellList with the cells which cannot be used by the neighbour. The rule to select the method is implementation-specific. 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 is not occupied and choose channelOffset randomly for each cell.

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 from the unallocated cells on the schedule, verifying that the slot offset is not occupied from the ones on the CellList.

SF0 does not include any transaction retry process. If the transaction is not successful, SF0 will be retriggered on the next slotframe if the number of used cells changes.

8. 6P Timeout Value

The timeout value is implementation-specific. The timeout value MAY be different for each transaction and each neighbour. The timeout range is from 0 to 128. The timeout MUST be added as an 7-bit on the Metadata header to the neighbour. There is no measurement unit associated to the timeout value. If the timeout expires, the node issues a RESET return code will be issued to the neighbour. SF0 has no retry policy. Timeout examples are depicted on Figure 3 and Figure 4.

```
|Timeout Value-----|
|A|-----First Exchange----->|B|-----Second Exchange----->|A|
|Complete transaction-----|
```

Figure 3: Example Transaction where the timeout does not expire


```
|Timeout Value-----|
|A|-----First Exchange----->|B|-----Second Exchange----->|A|
|Non-Complete transaction-----|
```

Figure 4: Example Transaction where the timeout expires

9. 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 [TIMEOUT] represents the Timeout value
 BIT 15 [WBLIST] is used to indicate that the CellList provided is
 a Whitelist (value=0) or a Blacklist (value=1).

10. 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 neighbor nodes to enable a new allocation process and at least a SF0THRESH number of cells MUST be allocated to each of the neighbours.

11. Cell Type

SF0 uses TX (Transmission) cell type only, thus defining celloptions as TX=0, RX=1 and S=0 according to section 4.2.6 of [\[I-D.ietf-6tisch-6top-protocol\]](#).

12. SF0 Statistics

Packet Delivery Rate (PDR) is calculated per cell, as the percentage of acknowledged packets, for the last 10 packet transmission attempts. There is no retransmission policy on SF0.

13. Relocating Cells

Allocated cells may experience packet loss from different sources, such as noise, interference or cell collision (after the same cell is allocated by other nodes in range on the network).

SF0 uses Packet Delivery Rate (PDR) statistics to monitor the currently allocated cells for cell relocation (by changing their slotOffset and/or channelOffset). When the PDR of one or more softcells is below PDR_THRESHOLD, SF0 relocates each of the cell(s) to a number of available cells selected randomly. PDR_THRESHOLD is out of the scope of this document and it is implementation-dependent.

14. Forced Cell Deletion Policy

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 neighbor list. The action to follow this condition is out of scope of this document and it is implementation-dependent.

15. 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 ADD 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 ADD Request, the neighbor has received the request, but less than NumCells of the cells in the CellList were allocated. 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 to a blacklist. The node MAY retry to contact this neighbor later.

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

RC_ERR_GEN: The node MUST issue a CLEAR command to the neighbour.

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

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

RC_ERR_RESET: Abort 6P Transaction

RC_ERR: Abort 6P Transaction. The node MAY retry to contact this neighbor later.

16. Experimental requirements

In order to evaluate the performance of this draft, we propose the following experimental work:

1. Define values for OVERPROVISION, SF0THRESH and ranges to the number of cells to Add or Delete after the Allocation Policy is applied for typical use cases.
2. Analyze the scheduling stability (in terms of oscillation) and the hysteresis effect on scheduling using SFX. A tradeoff shall be found between the reactivity of the algorithm facing new scheduling requirements and the number of overprovisioned cells.

3. Define the PDR value below the Average which is most effective for blacklisting cells and a method to whitelist cells. Analyze the stability and long-term behavior of this algorithm.
4. Measure the distribution of cell scheduling delay (including the time taken by 6P) to estimate timeouts for different type of transactions

17. Examples

TODO

18. 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.

19. Security Considerations

TODO

20. IANA Considerations

- o IANA_6TiSCH_SFID_SF0

21. 6P Compliance

- o MUST specify an identifier for that SF. OK
- o MUST specify the rule for a node to decide when to add/delete one or more cells to a neighbor. OK
- o MUST specify the rule for a Transaction source to select cells to add to the CellList field in the 6P ADD Request. OK
- o MUST specify the rule for a Transaction destination to select cells from CellList to add to its schedule. OK
- o MUST specify a value for the 6P Timeout, or a rule/equation to calculate it. OK
- o MUST specify a meaning for the "Metadata" field in the 6P ADD Request. OK
- o MUST specify the behavior of a node when it boots. OK
- 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). OK
- 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. OK
- o SHOULD clearly state the application domain the SF is created for. OK
- 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. OK

22. 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).

23. References

23.1. 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, <<https://www.rfc-editor.org/info/rfc2119>>.

23.2. Informative References

- [I-D.ietf-6tisch-6top-protocol]
Wang, Q., Vilajosana, X., and T. Watteyne, "6top Protocol (6P)", [draft-ietf-6tisch-6top-protocol-07](#) (work in progress), June 2017.
- [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-09](#) (work in progress), June 2017.
- [OpenWSN] Watteyne, T., Vilajosana, X., Kerkez, B., Chraïm, 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.
- [RFC6982] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", [RFC 6982](#), DOI 10.17487/RFC6982, July 2013, <<https://www.rfc-editor.org/info/rfc6982>>.

Appendix A. [TEMPORARY] Changelog

- o [draft-ietf-6tisch-6top-sf0-02](#)
 - * Editorial changes (figs, typos, ...)
- o [draft-ietf-6tisch-6top-sf0-03](#)
 - * Fixed typos
 - * Removed references to "effectively used cells"
 - * Changed Cell Estimation Algorithm to the third proposed alternative on IETF97
 - * Forced cell deletion becomes implementation specific
 - * Added PDR calculation formula
 - * Added PDR_THRESHOLD as implementation specific value

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
Luxembourg Institute of Science and Technology (LIST)
Department 'Environmental Research and Innovation' (ERIN)
41, rue du Brill
Belvaux L-4422
Grand-duchy of Luxembourg

Phone: +352 275 888-5055
Email: mariarita.palattella@list.lu

Nicola Accettura
LAAS-CNRS
7, avenue du Colonel Roche
Toulouse 31400
France

Phone: +33 5 61 33 69 76
Email: nicola.accettura@laas.fr

