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

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

October 19, 2015

6TiSCH 6top Scheduling Function Zero (SF0)

draft-dujovne-6tisch-6top-sf0-00

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 specific application's bandwidth requirements and the network condition. 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 will expire on April 21, 2016.

Copyright Notice

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 5
6. Meaning of Container Field 5
7. Node Behavior at Boot 5
8. Relocating Cells 6
9. 6P Error Handling 6
10. Examples 6
11. Implementation Status 6
12. Security Considerations 7
13. IANA Considerations 7
14. Acknowledgements 7
15. References 7
15.1. Normative References 7
15.2. Informative References 8
Authors' Addresses 8

1. Introduction

This document defines the a 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 for a particular neighbor to determine how many cells are 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 monitor the bandwidth usage on the node (local node bandwidth) and bandwidth requests from neighbour nodes (incoming bandwidth). 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 local bandwidth requirements. This allows the node to calculate the total outcoming bandwidth requirement. As a consequence, the Bandwidth Estimation Algorithm for SF0 follows the steps described below:

1. Collect the Incoming Bandwidth Requirements from neighbour nodes (IBR).
2. Collect the Local node Bandwidth Requirements (LBR).
3. Calculate the updated total Outgoing Bandwidth Requirement (OBR) as: OBR=LBR+IBR and submit the request to the allocation policy.
4. Return to step 1.

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 cell 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.

```
                SCHEDULEDCELLS
    <--------------------------------->
   +---+---+---+---+---+---+---+---+---+
   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+
                   |<----------------->|
                   |     SF0THRESH     |
                   |                   |
   REQUIREDCELLS   |                   |
   +---+---+       |                   |       DELETE
   |   |   |       |                   |       ONE/MORE
   +---+---+       |                   |       CELLS
                   |                   |
       REQUIREDCELLS                   |
   +---+---+---+---+---+---+           |       DO
   |   |   |   |   |   |   |           |       NOTHING
   +---+---+---+---+---+---+           |
                   |                   |
               REQUIREDCELLS           |
   +---+---+---+---+---+---+---+---+---+---+   ADD
   |   |   |   |   |   |   |   |   |   |   |   ONE/MORE
   +---+---+---+---+---+---+---+---+---+---+   CELLS
```

Figure 1: The SF0 Allocation Policy

1. If REQUIREDCELLS<(SCHEDULEDCELLS-SF0THRESH), delete one or more cells.
2. If (SCHEDULEDCELLS-SF0THRESH)<=REQUIREDCELLS<=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.

4. Rules for CellList

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

The Transaction Source node, for each of the cells to be put in the CellList field, first selects the slotOffset randomly; second, it verifies if the slotOffset is free and third it chooses the channelOffset randomly.

The Transaction Destination node goes through the cells in the CellList in order. For each one, it verifies whether it has a cell schedule with the same slotOffset. If yes, it skips the cell. If not, it allocates the cell. This stops when either (1) it has scheduled NumCells cells or (2) there are no more cells in 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: timeout = (2^(macMaxBE+1)-2^macMinBE) * SM

6. Meaning of Container Field

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.

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 is much lower than average.

9. 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.

10. Examples

TODO

11. 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.

12. Security Considerations

TODO

13. IANA Considerations

- IANA_SFID_SF0

14. 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).

15. References

15.1. 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) Amendament 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.

15.2. 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-05 (work in progress), July 2015.

[I-D.wang-6tisch-6top-sublayer]
Wang, Q. and X. Vilajosana, "6TiSCH Operation Sublayer (6top)", draft-wang-6tisch-6top-sublayer-02 (work in progress), October 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