

6TiSCH
Internet-Draft
Intended status: Standards Track
Expires: January 4, 2018

S. Duquennoy, Ed.
Inria
X. Vilajosana
Universitat Oberta de Catalunya
T. Watteyne
Inria
July 3, 2017

6TiSCH Autonomous Scheduling Function (ASF)
[draft-duquennoy-6tisch-asf-00](#)

Abstract

This document defines a Scheduling Function called "ASF": the 6TiSCH Autonomous Scheduling Function. With ASF, nodes maintain their TSCH schedule based on local neighborhood knowledge, without any signaling. Hashes of the nodes' MAC address are used to deterministically derive the [slotOffset, channelOffset] location of cells in the TSCH schedule. The MAC, control, and application traffic planes are assigned to distinct slotframes, for isolation and flexible dimensioning. This approach provides over-provisioned schedules with low maintenance, in pursuit for simplicity rather than optimality.

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 January 4, 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](#) 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
2.1.	Application Domains	3
3.	General Operation	4
3.1.	Types of Slotframes	4
3.2.	Cell Coordinates	4
3.3.	Slotframes Definition	5
4.	Configuration	5
5.	Scheduling Function Identifier	6
6.	Rules for Adding/Deleting Cells	6
7.	Rules for CellList	7
8.	6P Timeout Value	7
9.	Rule for Ordering Cells	7
10.	Meaning of the Metadata Field	7
11.	Node Behavior at Boot	8
12.	6P Error Handling	8
13.	Examples	8
14.	Implementation Status	8
15.	Security Considerations	9
16.	IANA Considerations	9
16.1.	6P Scheduling Function Identifiers 'ASF'	9
17.	References	9
17.1.	Normative References	9
17.2.	Informative References	9
Appendix A.	Contributors	10
Appendix B.	Acknowledgments	10
Appendix C.	[TEMPORARY] Changelog	10
	Authors' Addresses	11

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, etc). The statement does indicate that the text will be written at some point.
- 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 "RFCXXXX" refers to the RFC number of this specification, once published.
- 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, and will not be part of the final text.
- o This section will be removed in the final text.

2. Introduction

This document defines an autonomous Scheduling Function for the 6top sublayer [[I-D.ietf-6tisch-6top-protocol](#)], called "ASF". It is designed to operate without any signaling, keeping the TSCH schedule consistent between neighbors at all times (matching slots for transmission and reception). ASF uses 6P for neighbor schedule inspection (commands STATUS and LIST), but not for adding/deleting cells. ASF isolates the traffic from different planes in distinct slotframes, so as to avoid any disruption between MAC synchronization, control and application traffic.

ASF addresses all requirements listed in Section "Requirements for an SF" from [[I-D.ietf-6tisch-6top-protocol](#)]. The organization of this document follows section "Recommended Structure of an SF Specification" in [[I-D.ietf-6tisch-6top-protocol](#)]. This document follows the terminology defined in [[I-D.ietf-6tisch-terminology](#)].

2.1. Application Domains

ASF is primarily targeted at applications with random traffic flows, such as interactive CoAP traffic. Its main strength is its signaling-free nature, which ensures the slots installed at neighboring nodes are consistent at all times. Its main weakness is its contention-based nature and its need to over-provision the

schedule, rendering it unable to meet stringent latency and energy requirements. An example application domains is building instrumentation. ASF was evaluated experimentally and shown to achieve over 99.99% end-to-end delivery in 6TiSCH/RPL testbeds [[Orchestra-SenSys](#)].

3. General Operation

ASF uses multiple slotframes, each assigned to one particular type of traffic, e.g. TSCH synchronization, routing control or application traffic. Nodes maintain the cells within the slotframes autonomously, based on the hash of either the source's or destination's MAC address. Each slotframe is uniquely assigned a set of channel offsets.

3.1. Types of Slotframes

There are three different types of slotframes, described next:

Rendez-vous slotframe: Contains a contention-based rendez-vous cell with fixed coordinates, options Rx, Tx, and Shared. The cell can be used for any type of traffic, including broadcast. This slotframe is equivalent to the 6TiSCH minimal schedule [[RFC8180](#)].

Receiver-based slotframe: Nodes have one receive (option Rx) cell at coordinates derived from a hash of their MAC address. For transmitting to a given neighbor, nodes maintain a cell (options Tx, Shared) at coordinates derived from a hash of the neighbor's MAC address. For instance, a node may maintain such a transmit cell for each known neighbor in the IPv6 NDP cache, or to a subset such as the RPL preferred parent(s) or TSCH time source(s).

Sender-based slotframe: Nodes have one transmit cell (options Tx, Shared) at coordinates derived from a hash of their MAC address. To listen to a given neighbor, they maintain a cell (option Rx) at coordinates derived from a hash of the neighbor's MAC address. Typically, nodes only listen to a subset of neighbors, such as the RPL preferred parent(s) or TSCH time source(s).

3.2. Cell Coordinates

Cell coordinates in ASF are either fixed (for rendez-vous slotframes) or derived from a MAC address (for receiver- and sender-based slotframes). To derive coordinates from a MAC address, nodes MUST use the hash function SAX [[SAX-DASFAA](#)] on the EUI-64. Let S_len be the length of slotframe S , and $S_channels$ be the set of channels assigned to slotframe S . The slot coordinates derived from a given MAC address are computed as follows:

$$slotOffset(MAC) = hash(MAC) \% S_len$$


```
channelOffset(MAC) = S_channels[(hash(MAC) / L) % len(S_channels)]
```

3.3. Slotframes Definition

By default, nodes maintain the four following slotframes:

SlotframeA: TSCH Enhanced Beacons: Nodes MUST send all TSCH Enhanced Beacons (EBs) on this slotframe. This is a sender-based slotframe. Nodes have one transmit cell based on their MAC address, and listen to their TSCH time source with one receive cell (with additional option TimeKeeping).

SlotframeB: TSCH Keep-Alives: Nodes MUST send all Keep-Alives (KAs) on this slotframe. This is a receiver-based slotframe. Nodes have one receive cell based on their MAC address, and one transmit cell towards their TSCH time source (with additional option TimeKeeping).

SlotframeC: Application traffic: Nodes MUST send all unicast UDP and TCP frames in this slotframe. This is a received-based slotframe. Nodes have one receive cell based on their MAC address, and maintain as many transmit cells as they have neighbors in their IPv6 NDP cache. Every time a node is added/removed to the NDP cache, a new transmit cell is immediatly added/removed.

SlotframeD: Other Traffic: Nodes MUST send all other traffic to this slotframe, such as 6P or ICMPv6 traffic. This is a rendez-vous slotframe. All nodes have one single cell at slotOffset 0 and at the first channel offset of the slotframe.

As the slotframes repeat over time, cells from different slotframes will overlap periodically. In case a node has multiple cells schedule at the same time, the precedence rules from [\[IEEE802154-2015\]](#) apply.

In order to fully isolate the different planes, we RECOMMEND provisioning separate packet queues for each slotframe. This ensures that transient bursts at the application layer will not affect TSCH synchronization nor routing topology maintenance. Conversely, occasional routing or TSCH traffic will not affect each other nor affect the application.

4. Configuration

ASF defines a set of configuration parameters listed in Figure 1. In order to distribute cell overlap uniformly (see [Section 3.3](#)), we RECOMMEND selecting slotframe lengths that are co-primes. TODO define how configuration is discovered. Can be through new EB IEs, or through new/extended 6P command(s).

Property	Example Setting
SlotframeA length	397
SlotframeA handle	4 (lowest precedence)
SlotframeA channelOffsets	0
SlotframeA type of cells	ADVERTISING
SlotframeB length	389
SlotframeB handle	0 (highest precedence)
SlotframeB channelOffsets	1
SlotframeB type of cells	NORMAL
SlotframeC length	Trades-off app. traffic capacity against energy. Ex. value: 17
SlotframeC handle	1
SlotframeC channelOffsets	2 to 14
SlotframeA type of cells	NORMAL
SlotframeD length	31
SlotframeD handle	2
SlotframeD channelOffsets	15
SlotframeD type of cells	NORMAL

Figure 1: Example Settings for ASF.

5. Scheduling Function Identifier

The Scheduling Function Identifier (SFID) of ASF is IANA_SFID_ASF.

6. Rules for Adding/Deleting Cells

ASF nodes maintain their cells autonomously, and do not use 6P ADD nor DELETE.

7. Rules for CellList

For the 6P LIST command, ASF uses the default CellList field format defined in [Section 4.2.4](#) [TODO: update if needed] of [\[I-D.ietf-6tisch-6top-protocol\]](#).

8. 6P Timeout Value

The timeout is of low criticality in ASF as 6P Requests are only used for schedule inspection, not for cell addition/removal. The RECOMMENDED timeout value in slots is:

$$2^{(\text{macMaxBe}+2)} \times \text{SlotframeD_len}$$

which is an upper bound of the maximum time spent in transmission attempts of a 6P Request and Response, over slotframeD (where 6P traffic is sent). The upper bound is conservative, giving extra time for time spent in packet queues.

Assuming default values ([[IEEE802154-2015](#)] and Figure 1), that is $\text{macMaxBe}=5$ and $\text{Slotframe2_len}=31$, this results in a timeout of 3968 timeslots.

9. Rule for Ordering Cells

Cells are ordered by increasing slotframe handle, then by timeslot, then channel offset.

10. Meaning of the Metadata Field

The Metadata 16-bit field is used as follows: Figure Figure 2 shows the format of the Metadata field, where:

- o Slotframe: is used to identify a slotframe by its handle.
- o BITS 8-15 are reserved.

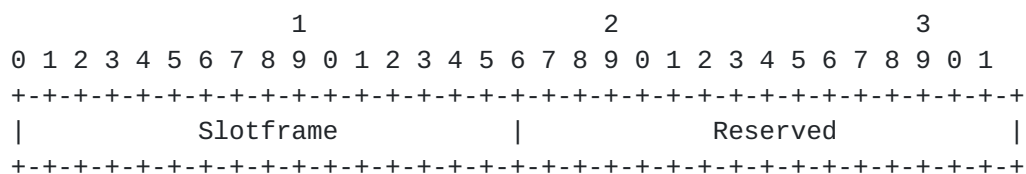


Figure 2: Format of the Metadata Field.

11. Node Behavior at Boot

At boot, ASF creates four empty slotframes with length and handle described in [Section 4](#).

TODO describe configuration discovery.

12. 6P Error Handling

ASF only uses 6P commands COUNT and LIST. In case of error on STATUS or LIST, the node MAY retry to contact this neighbor after the 6P timeout.

13. Examples

TODO

14. 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".

Contiki: The mechanism behind this specification is implemented in the Contiki project [\[Contiki\]](#). Adjustments to exactly match this specification are in progress. The mechanism was evaluated experimentally in large-scale testbeds in [\[Orchestra-SenSys\]](#).

15. Security Considerations

ASF is not threatened by attacks on 6P messages as it operates without signaling. However, it bases its TSCH schedule on external information, namely: (1) the identify of the current TSCH time source and (2) the MAC address of its neighbors. ASF relies on link-layer security to ensure the integrity of the above information.

16. IANA Considerations

16.1. 6P Scheduling Function Identifiers 'ASF'

This document adds the following number to the "6P Scheduling Function Identifiers" registry defined by [\[I-D.ietf-6tisch-6top-protocol\]](#):

SFID	Name	Reference
IANA_6TiSCH_SFID_ASF	Autonomous Scheduling Function (ASF)	TODO

Figure 3: 6P Scheduling Function Identifiers 'ASF'.

17. References

17.1. Normative References

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

17.2. Informative References

- [Contiki] Dunkels, A., Lignan, A., Thebaudeau, B., Quattlebaum, R., Rosendal, F., Oikonomou, G., Deru, L., Alvira, M., Tsiftes, N., Schmidt, O., and S. Duquennoy, "The Contiki Open Source OS for the Internet of Things", <https://github.com/contiki-os/contiki>, November 2016.

[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.

[Orchestra-SenSys]

Duquennoy, S., Al Nahas, B., Landsiedel, O., and T. Watteyne, "Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH", ACM SenSys 2015 , November 2015.

[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>>.

[RFC8180] Vilajosana, X., Ed., Pister, K., and T. Watteyne, "Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e (6TiSCH) Configuration", [BCP 210](#), [RFC 8180](#), DOI 10.17487/RFC8180, May 2017, <<http://www.rfc-editor.org/info/rfc8180>>.

[SAX-DASFAA]

Ramakrishna, M. and J. Zobel, "Performance in Practice of String Hashing Functions", DASFAA , 1997.

[Appendix A. Contributors](#)

Beshr Al Nahas (Chalmers University, beshr@chalmers.se) and Olaf Landsiedel (Chalmers University, olafl@chalmers.se) contributed to the design and evaluation of ASF.

[Appendix B. Acknowledgments](#)

TODO people

TODO projects

[Appendix C. \[TEMPORARY\] Changelog](#)

o [draft-duquennoy-6tisch-asf-00](#)

* Initial draft.

Authors' Addresses

Simon Duquennoy (editor)
Inria
40 Avenue Halley
Building A
Villeneuve d'Ascq
France

Email: simon.duquennoy@inria.fr

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

Email: xvilajosana@uoc.edu

Thomas Watteyne
Inria
2 Rue Simone Iff
Paris
France

Email: thomas.watteyne@inria.fr

