6TiSCH S. Duquennoy, Ed.
Internet-Draft RISE SICS
Intended status: Standards Track X. Vilajosana
Expires: September 2, 2018 Universitat Oberta de Catalunya
T. Watteyne
Inria
March 1, 2018

6TiSCH Autonomous Scheduling Function (ASF)
draft-duquennoy-6tisch-asf-01

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 after association. Hashes of the nodes' MAC address are used to deterministically derive the [slotOffset,channelOffset] location of cells in the TSCH schedule. Different traffic types (e.g. TSCH EB, RPL DIO, UDP etc.) 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 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 September 2, 2018.

Copyright Notice

Copyright (c) 2018 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 (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
   2.1. Application Domains 3
3. General Operation 4
   3.1. Cell Coordinates 4
   3.2. Types of Slotframes 4
      3.2.1. Rendez-vous slotframe 4
      3.2.2. Receiver-based slotframe 4
      3.2.3. Sender-based slotframe 5
   3.3. Conditional Cells 5
   3.4. Interaction between Slotframes 5
4. Configuration 6
5. Scheduling Function Identifier 9
6. Rules for Adding/Deleting Cells 9
7. Rules for CellList 9
8. 6P Timeout Value 10
9. Rule for Ordering Cells 10
10. Meaning of the Metadata Field 10
11. Node Behavior at Boot 10
12. 6P Error Handling 10
13. Examples 11
14. [TEMPORARY] Implementation Status 11
15. Security Considerations 11
16. IANA Considerations 12
   16.1. 6P Scheduling Function Identifiers 'ASF' 12
17. References 12
   17.1. Normative References 12
   17.2. Informative References 12
Appendix A. Contributors 13
Appendix B. Acknowledgments 13
Appendix C. [TEMPORARY] Changelog 13

Authors' Addresses 14

1. TEMPORARY EDITORIAL NOTES

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

- "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.
- "TEMPORARY" appendices are there to capture current ongoing discussions, or the changelog of the document. These appendices will be removed in the final text.
- "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.
- "RFCXXXX" refers to the RFC number of this specification, once published.
- 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.
- 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 runtime signaling, keeping the TSCH schedule consistent between neighbors at all times (slots for transmission and reception always match). ASF uses 6P solely for configuration at association time (6P SIGNAL) and for schedule inspection (6P STATUS and LIST). ASF isolates different traffic types into 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 EBs, RPL or UDP 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. Cell Coordinates

Cell coordinates in ASF are either fixed or derived from a MAC address (depending on the slotframe type, see Section 3.2). To derive coordinates from a MAC (EUI-64) address, nodes MUST use the hash function provided at configuration time, see Section 4. One example hash function is SAX [SAX-DASFAA]. 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)]
```

3.2. Types of Slotframes

There are three different types of slotframes, described next. Section 4 provides full details on cell options and other aspects.

3.2.1. Rendez-vous slotframe

Contains a single contention-based rendez-vous cell, at coordinates [slot offset: 0; channel offset: 0]. This slotframe is equivalent to the 6TiSCH minimal schedule [RFC8180].

3.2.2. Receiver-based slotframe

One Rx cell: Coordinates computed as the hash of the node's own MAC address.

Multiple Tx cells: One Tx cell per neighbor. Coordinates computed as the hash of the neighbor's MAC address.

3.2.3. Sender-based slotframe

One Tx cell: Coordinates computed as the hash of the node's own MAC address.

Multiple Rx cells: One Rx cell per neighbor. Coordinates computed as the hash of the neighbor's MAC address.

3.3. Conditional Cells

In order to handle traffic bursts, ASF utilizes conditional cells. When a node has several frames in its queue for a given neighbor, it can set the [IEEE802154-2015] 'frame pending' bit in unicast transmissions to that neighbor. Cells at upcoming time offsets will be used to carry more frames. Note that collisions may happen on these conditional cells, which MUST therefore have the 'Shared' bit set.

Sender: A sender with multiple unicast frames in its queue for a given neighbor MAY send frames with the 'frame pending' bit set. After sending a unicast frame with the 'frame pending' bit set, if a link-layer Acknowledgment (ACK) is received, the sender immediately schedules a temporary Tx cell. Compared to the initial cell, the temporary cell has the same Link Options plus the 'Shared' bit, the same channel offset, and a time offset incremented by 1 (modulo the slotframe length). The next frame will be sent on this temporary cell, and may set the 'frame pending' bit again to signal more traffic to come. The procedure repeats until the transmit queue is empty, or until no acknowledgment is received for a frame.

Receiver: Upon receiving a unicast frame with the 'frame pending' bit set, the node first sends a link-layer ACK. It then schedules a temporary Rx cell, with same Link Options, same channel offset, and time offset incremented by 1. If, in the new cell, it receives a unicast frame with the 'frame pending' bit set, it continues scheduling additional Rx cells to receive subsequent frames.

3.4. Interaction between Slotframes

ASF is expected to maintain multiple slotframes, each dedicated to a different traffic type. As the slotframes repeat over time, cells from different slotframes overlap periodically. In case a node has multiple cells scheduled at the same time, the precedence rules from [IEEE802154-2015] apply. In order to distribute cell overlap uniformly, it is RECOMMENDED to select slotframe lengths that are co-primes.

4. Configuration

An ASF configuration consists of a series of slotframes with attributes. ASF uses the 6P SIGNAL command (format in Figure 1) to disseminate its configuration. SIGNAL commands are directly included as IETF IE in each EB, so that nodes learn the ASF configuration directly at join-time.

6TiSCH EBs are not secured. For applications that require the ASF schedule to be sent securely, the ASF SIGNAL command MAY be sent instead in separate data broadcast packets, after join-time. To summarize, there are two cases:

Initial EB includes ASF SIGNAL: The node configures itself to run the ASF configuration provided.

Initial EB does not include ASF SIGNAL: The node runs the 6TiSCH minimal schedule ([RFC8180]) at association. When later receiving a packet with ASF SIGNAL, the node replaces the 6TiSCH minimal schedule with the ASF configuration provided.

Figure 1 describes the format of the ASF SIGNAL command.

```
                  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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| # Slotframes  | Slotframe list ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Format of the ASF SIGNAL command.

Where:

Slotframes: The number of ASF slotframes

Slotframe list: The list of slotframes, with format described in Figure 2

```
                    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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Slotf. handle |         Slotframe size        |      Type     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Min. channel offset      |      Max. channel offset      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Tx Cell Opt  |  Rx Cell Opt  |    Nbr Set    |   Hash func.  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   # Filters   | Traffic filter list ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Figure 2: Format of the ASF SIGNAL slotframe descriptor.

Figure 2 shows the format of a slotframe descriptor, where:

Slotf. handle: The IEEE 802.15.4 slotframe handle (8 bits)
Slotframe size: The IEEE 802.15.4 slotframe size (16 bits)
Type: The ASF slotframe type. The set of possible values for this field is presented in Figure 3. The different slotframe types are described in Section 3.2.
Min. channel offset: ASF slotframes are assigned a channel offset range. This defines the lower bound for the range.
Max. channel offset: ASF slotframes are assigned a channel offset range. This defines the upper bound for the range.
Tx Cell Opt: The options to be used for the Tx Cells, if any.
Rx Cell Opt: The options to be used for the Rx Cells, if any.
Nbr Set: The set of neighbors for which Cells are instantiated. The set of possible values for this field is presented in Figure 4.
Hash func.: The hash function used to compute cell coordinates from a node's EUI-64 address, as defined in Section 3.1. The set of possible values for this field is presented in Figure 5.
Filters: ASF slotframes are assigned a subset of the traffic each. One or several traffic filters will be applied to only select packets with the intended properties. When there are several traffic filters, they are combined with a OR, i.e., packets that satisfy any of the filters will be sent on the slotframe. This field defines how many filters are in place. The filter descriptions follow inline, with format defined in Figure 6.

Num.	Description
0	Rendez-vous slotframe
1	Receiver-based slotframe
2	Sender-based slotframe
128--255	Reserved

Figure 3: Field: types of slotframes.

Num.	Description
0	Empty set
1	All TSCH time sources
2	All RPL parents
3	The RPL preferred parent
4	All IPv6 NDP neighbors
128--255	Reserved

Figure 4: Field: neighbor set.

Num.	Description
0	SAX (Shift-Add-XOR)
1--255	Reserved

Figure 5: Field: Hash function

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   FT   | UC/BC |  IP protocol  |      Type / Code / Port       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Format of the ASF SIGNAL traffic filters.

The fields for traffic filter descriptions (Figure 6) are described next:

FT: The IEEE 802.15.4 frame type. Examples; 0: Beacon; 1: Data.

UC/BC: Unicast/broadcast nature. Possible values; 0: unicast; 1: broadcast; 2: any.

IP protocol: The IP protocol number. Examples; 0x3a: ICMPv6; 0x11: UDP. A value of 0x00 ignores this field.

Type / Code / Port: In the case of ICMPv6: the Type (first 8 bits) followed by the Code (last 8 bits). In case of UDP or TCP: the 16-bit port. A value of 0x00 ignores this field.

Example filters are given next:

All TSCH EBs: FT: 0 (beacon), UC/BC: 1 (broadcast), IP protocol: 0, Type: 0.

All RPL Traffic: FT: 1 (data), UC/BC: 2 (unicast and broadcast), IP protocol: 0x3a (ICMPv6), Type: 0x9b (RPL), Code: 0x00

RPL Unicast DIO: FT: 1 (data), UC/BC: 0 (unicast), IP protocol: 0x3a (ICMPv6), Type: 0x9b (RPL), Code: 0x01 (DIO)

UDP port 5683: FT: 1 (data), UC/BC: 2 (unicast and broadcast), IP protocol: 0x11 (UDP), Port: 5683

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^(macMaxBe+2)*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.

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 7 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.

```
                  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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Slotframe           |            Reserved           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: Format of the Metadata Field.

11. Node Behavior at Boot

At boot, nodes start with an empty schedule. When associating, they configure their schedule with the 6P ASF SIGNAL command, which is included either in the initial EB or later packets, as described in Section 4.

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. [TEMPORARY] 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

At run-time, 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.

At configuration time, ASF relies on a 6P SIGNAL command. This command MAY be secured as described in Section 4. When this command is not secured, the security of the network is equivalent to that of the 6TiSCH minimal configuration ([RFC8180]). That is, the network schedule is propagated directly through EBs.

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)	RFCXXXX

Figure 8: 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, <https://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-09 (work in progress), October 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, <https://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, <https://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-01

   * Defines ASF configuration parameters and procedure;
   * Defines packet format to disseminate configurations (6P signal);
   * Defines burst mode (conditional cells based on 'frame pending' bit);
   * Makes Hash function configurable (SAX remains default).

o draft-duquennoy-6tisch-asf-00

  * Initial draft.

Authors' Addresses

Simon Duquennoy (editor)
RISE SICS
Isafjordsgatan 22
164 29 Kista
Sweden

Email: simon.duquennoy@ri.se

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