6TiSCH
Internet-Draft
Intended status: Informational
Expires: January 5, 2015

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

July 4, 2014

6TiSCH On-the-Fly Scheduling
draft-dujovne-6tisch-on-the-fly-03

Abstract

This document describes the environment, problem statement, and goals of On-The-Fly (OTF) scheduling, a Layer-3 mechanism for 6TiSCH networks. The purpose of OTF is to dynamically adapt the aggregate bandwidth, i.e., the number of reserved soft cells between neighbor nodes, based on the specific application constraints to be satisfied. When using OTF, softcell and bundle reservation is distributed: through the 6top interface, neighbor nodes negotiate the cell(s) to be (re)allocated/deleted, with no intervention needed of a centralized entity. This document aims at defining a module which uses the functionalities provided by the 6top sublayer to (i) extract statistics and (ii) determine when to reserve/delete soft cells in the schedule. The exact reservation and deletion algorithm, and the number and type of statistics to be used in the algorithm are out of scope. OTF deals only with the number of softcells to be reserved/deleted; it is up to 6top to select the specific soft cells within the TSCH schedule.

This Internet-Draft will expire on January 5, 2015.

Copyright Notice

Table of Contents

1. Introduction

The IEEE802.15.4e standard [IEEE802154e] was published in 2012 as an amendment to the Medium Access Control (MAC) protocol defined by the IEEE802.15.4-2011 [IEEE802154] standard. The Timeslotted Channel Hopping (TSCH) mode of IEEE802.15.4e is the object of this document.

On-The-Fly (OTF) scheduling is a 1-hop protocol with which a node negotiates the number of soft cells scheduled with its neighbors,

without requiring any intervention of a centralized entity (e.g., a PCE). This document describes the OTF allocation policies and methods used by two neighbors to allocate one or more softcells in a distribution fashion. It also proposes an algorithms for estimating the required bandwidth (BW). This document defines the interface

between OTF and the 6top sublayer ([I-D.wang-6tisch-6top]), to collect and retrieve statistics, or allocate/delete cells and bundles. This document defines a framework; the algorithm and statistics used are out of scope. This draft follows the terminology defined in [I-D.ietf-6tisch-terminology] and addresses the open issue related to the scheduling mechanisms raised in [I-D.ietf-6tisch-tsch].

2. Allocation policy

OTF is a distributed scheduling protocol which increases/decreases the bandwidth between two neighbor nodes (i.e., adding/deleting sofcells) by interacting with the 6top sublayer. It retrieves statistics from 6top, and uses that information to trigger 6top to add/delete softcells to a particular neighbor. The algorithm which decides when to add/delete softcells is out of scope. For example, 6top might decide to add a cell if some queue of outbound frames is overflowing. Similarly, OTF can delete cells when the queue has been empty for some time. OTF only triggers 6top to add/delete the soft cells, it is the responsibility of the 6top sublayer to determine the exact slotOffset/channelOffset of those cells. In this document, the term "cell" and "soft cell" are used interchageably.

All the softcells allocated are part of best effort track, i.e. with TrackID=00, as defined i [I-D.wang-6tisch-6top]. These cells can be used for forwarding any packet in the queue, regardless of the specific track it belongs to. OTF manages the global bandwidth requirements between two neighbor nodes; per-track management is currently out of scope.

OTF is prone to schedule collisions. Nodes might not be aware of the cells allocated by other pairs of nodes. A schedule collision occurs when the same cell is allocated by different pairs in the same interference space. The probability of having allocation collision may be kept low by grouping cells into chunks (see [I-D.ietf-6tisch-terminology] and [I-D.ietf-6tisch-architecture] for

more details). The use of chunks is outside the scope of this current version of the OTF draft.

The "allocation policy" is the algorithm used by OTF to decide when to increase/decrease the bandwidth allocated between two neighbor nodes in order to satisfy the traffic requirements. These requirements can be expressed in terms of throughput, latency or other constraints.

An OTF allocation policy MAY be defined according to a combination of two different approaches: reactive and proactive.

In a reactive approach, the allocation policy follows the increased/decreased need for bandwidth. Upon reception of a bandwidth request, OTF sends softcell allocation requests to the 6top sublayer. OTF estimates the number of cells to be allocated per neighbor. If the traffic exchanged between two neighbors reduces, OTF asks 6top to de-allocate one or more cells. Once the cells are deleted, 6top notifies OTF, which updates its internal state.

In a proactive approach, the allocation policy over-provisions the number of cells reserved in a bundle between two neighbors, i.e., cells are scheduled in advance. When OTF issues a bundle allocation request to 6top, it indicates the desired size of the bundle and the TrackID=00. 6top selects the cells belonging to the bundle on the best effort track. Based on the network traffic conditions (e.g., queue utilization), some portion of those cells are used for communication. In any case, allocated cells within a bundle are consecutive, starting from the first cell in the block. The cells which are not currently used, are still reserved for that pair of nodes, for possible future use.

It is up to the implementor to select the approach most appropriate for the application. The reactive approach is, in general, be more energy-efficient (it allocates only the cells needed), at the expense of increased cell allocation latency (negotiating to add/delete cells takes some time).

The proactive approach compared to the reactive one reduces the cell allocation latency. Cells within a bundle are over-provisioned, and a priori scheduled. When needed, the 6top sublayer of the node can

allocate them, without going through any negotiation phase with the 6top layer of the neighbor node. Thus, the proactive approach provides a low-delay response after a surge in bandwidth usage. In fact, soft cells within a bundle are already scheduled and become immediately available, upon bandwidth request, without the need of a negotiation phase. The use of bundles does force the receiver module of the node to be active during the whole length of the bundle, resulting in increased power consumption.

This document introduces the following parameters to accomplish both approaches previously described:

SCHEDULEBW: The amount of cells scheduled in a bundle on the best effort track between two neighbors.

REQUIREDBW: Bandwidth requested by OTF to 6top, a non-negative number. How this is computed is out of the scope. It MAY be the an instantaneous bandwidth request, or a value averaged on several measurement, or an over-provisioned value.

PROACTIVETHRESH: Threshold parameter to introduce pro-activity in the allocation policy, described below. It is a non-negative bandwidth value. What value to use is application specific and out of scope. The maximum acceptable value for this parameter is equal to the current SCHEDULEBW.

The OTF allocation policy compares the required bandwidth against the scheduled one, using the pro-activity threshold for bounding the signaling overhead due to negotiations of new cells. In details:

1. If REQUIREDBW is greater than SCHEDULEBW, OTF asks 6top to add REQUIREDBW-SCHEDULEBW cells to the bundle on the best effort track.

2. If REQUIREDBW is greater or equal than SCHEDULEBW-PROACTIVETHRESH, and it is lower than or equal to SCHEDULEBW, OTF does not perform any bundle resizing, since the scheduled bandwidth is sufficient for managing the current traffic conditions.

3. If REQUIREDBW is lower than SCHEDULEBW-PROACTIVETHRESH, OTF asks 6top to delete SCHEDULEBW-PROACTIVETHRESH-REQUIREDBW from the

bundle on the best effor track.

A purely reactive approach uses PROACTIVETHRESH=0. In this case, OTF does not perform any allocating/deallocating operation when the required bandwidth is equal to the scheduled one.

A purely proactive approach uses PROACTIVETHRESH=SCHEDULEBW. In this case, OTF resizes the bundle only when the required bandwidth is greater than the scheduled one.

3. Allocation methods

Beyond the allocation policies that describe the approach used by OTF for fulfilling the node bandwidth requests, the OTF framework also includes Allocation Methods that specify how OTF issues commands to the 6top sublayer. In other words, the allocation methods represent the mechanisms that are used by the allocation policies.

In detail, OTF includes two distinct allocation methods: soft cell and bundle allocation methods. Each Allocation Policy can use either one or both allocation methods. As specified in [I-D.wang-6tisch-6top], 6top provides a set of commands that allows OTF to allocate/delete soft cells. The same set of commands can be used for reserving bundles.

With the soft cell allocation method, OTF has 6top reserve a single soft cell on the best effort track, for allowing a given node to exchange traffic with a specific neighbor. The 6top layer allocates and maintains this cell. If a bundle is already reserved between the same pair of neighbors, on the same track, this request translates into a bundle resize request. The newly allocated cell increase the size of the already existing bundle. Similarly, when OTF realizes there is a reduction of traffic exchanged between the two neighbors, it may asks 6top to delete a softcell from the best effort track, i.e. to decrease the size of the bundle on the best effort track. If no bundle with TrackID=00 exist, the 6top softcell create command generates a new bundle of size 1.

With the bundle allocation method, OTF sends bundle allocation requests to 6top sublayer, specifying the bundle size (the number of

soft cells) and the TrackID=00. Scheduling N softcells is equivalent to asking for a bundle of size N. The cells within the bundle are allocated by 6top (and thus, used for traffic exchange) only afterwards, according to the nodes bandwidth need.

4. Cell and Bundle Reservation/Deletion

In order to reserve/delete softcells, OTF interacts with 6top sublayer. To this aim OTF uses the following set of commands offered by 6top: CREATE.softcell, and DELETE.softcell. When creating (deleting) a softcell, OTF specified the track the cell belongs to (i.e., best effort track, TrackID=00), but not its slotOffset and channelOffset. If at least one cell on the best effort track already exists, the CREATE.softcell and DELETE.softcell, translate into INCREASE and DECREASE the bundle size, respectively. 6top is responsible for picking the specific cell to be added/deleted within the bundle. Before being able to do so, source and destination nodes go through a cell negotiation process. This process is out of scope of 6top and OTF. In order to reserve a best effort bundle, OTF uses the CREATE.softcell command, set TrackID=00, but asks 6top for multiple softcells. Following OTF request, 6top either (i) creates a new bundle, if no cells were reserved already on the best effort track, or (ii) increases the bundle size of the already existing best-effort bundle. By using the DELETE.softcell command, and asking for deleting multiple softcells, OTF has 6top delete the entire best effort bundle.

OTF provides a policy for 6top to generate CREATE/DELETE.softcells commands, policy that is out of 6top scope [I-D.wang-6tisch-6top]. Such policy is not the only one that can be used by 6top. Others may be defined in the future.

5. Getting statistics and other information about cells through 6top

Statistics are kept in 4 data structure of 6top MIB: CellList, MonitoringStatusList, NeighborList, and QueueList.

CellList provides per-cell statistics. From this list, an upper layer can get per-bundle statistics. OTF may have access to the CellList, by using the CoAP-YANG Model, but actually cell-specific

statistics are not significant to OTF, since softcells can be re-allocated in time by 6top itself, based on network conditions.

MonitoringStatusList provides per-neighbor and slotframe statistics. From it an upper layer (e.g., OTF) can get per bundle overview of scheduling and its performance. Such list contains information about the number of hard and soft cells reserved to a given node with a specific neighbor, and the QoS (that can be expressed in form of different metrics: PDR, ETX, RSSI, LQI) on the actual bandwidth, and the over-provisioned bandwidth (which includes the over-provisioned cells). 6top can use such list to operate 6top Monitoring Functions, such as re-allocating cells (by changing their slotOffset and/or channelOffset) when it finds out that the link quality of some softcell is much lower than average. Unlike 6top, OTF does not operate any re-allocation of cells. In fact, OTF can ask for more/ less bandwidth, but cannot move any cell within the schedule. Thus, the 6top Monitoring function is useful to OTF, because it can provide better cells for a given bandwidth requirement, specified by OTF. For instance, OTF may require some additional bandwidth (e.g. 2 cells in a specific slotframe) with PDR = 75%; then, 6top will reserve 3 slots in the slotframe to meet the bandwidth requirement. In addition, when the link quality drop to 50%, 6top will reserve 4 slots to keep meeting the bandwidth requirement. Given that OTF operates on the global bandwidth between two neighbor nodes, it does not need to be informed from 6top about cells' re-allocation.

NeighborList provides per-neighbor statistics. From it, an upper layer can understand the connectivity of a pair of nodes. Based on the quality of the link, e.g., LQI under threshold, OTF may ask 6top to delete some cells, in order to reserve them for better-connected links.

QueueList provides per-Queue statistics. From it, an upper layer can know the traffic load. OTF, based on such queue statistics (e.g., average length of the queue, average age of the packet in queue, etc.) may trigger a 6top CREATE.softcell (DELETEsoftcell) command for increasing (decreasing) the bandwidth and be able to better serve the packets in the queue.

6. Events triggering algorithms in OTF

The Algorithms running within OTF MUST be event-oriented. As a consequence, OTF requires to connect the algorithms with external events to trigger their execution. The algorithm also generates one or more events when it is executed, such as a new softcell allocation. Both type of events, the one which trigger the algorithm and the ones which are generated by the execution of the algorithm are called OTF events.

The following notation is used on the definition of OTF events:

BW <- BWA(B,T,S(T)) where:

BWA: Bandwidth allocation algorithm

BW: Bandwidth

T: Best Effort Track

B: Bundle

S(B,T) Statistics for bundle B on track T

M(B,T): Actual bundle size for bundle B on track T

The OTF events are defined as:

Event A: A new bundle B on track T is created. The OTF events generated by the algorithm are:

1. Add a new entry in the storage M for bundle B on track T.

2. Ask 6top for S(B,T).

3. BW <- BWA(B,T,S(T)).

4. Ask 6top to allocate a bundle of size BW.

5. M(B,T)<-BW.

Event B: A packet is waiting to be transmitted on any track, but no cell is available (i.e., saturation). The OTF events generated by the algorithm are:

1. Collect stats S from 6top.

2. BW <- BWA(B,T,S(T))).

3. Ask 6top to increase the bundle size up to BW.

4. If (allocation successful) then M(B,T)<-BW.

Event C: The usage of a bundle B on track T is too low, below a pre-established threshold. The OTF events generated by the algorithm are:

1. Collect stats S from 6top.

2. BW <- BWA(B,T,S(T)).

3. Ask 6top to decrease the bundle size to BW.

4. If (allocation successful) then M(B,T)<-BW.

Event D: The usage of a bundle B on track T is too high, above a pre-established threshold. The OTF events generated by the algorithm are:

1. Collect stats S from 6top.

2. BW <- BWA(B,T,S(T)).

3. Ask 6top to increase the bundle size to BW.

4. If (allocation successful) then M(B,T)<-BW.

Event E: Bundle B on track T is deleted. The OTF events generated by the algorithm are:

1. purge M(B,T).

7. Bandwidth Estimation Algorithms

OTF supports different bandwidth estimation algorithms that can be used by a node in a 6TiSCH network for checking the current traffic condition and thus the actual bandwidth usage. By doing so, one can adapt (increase or increase) the number of scheduled cells/bundles for a given pair of neighbors (e.g., parent node and its child), according to their needs. OTF supports several bandwidth estimation algorithms numbered 0 to 255 in the OTF implementation. The first algorithm (0) is reserved to the default algorithm that is described below. By using SET and GET commands, one can set the specific algorithm to be used, and get information about which algorithm is implemented.

Steps of the default bandwidth estimation algorithm, running over a parent node:

Step 1: Collect the bandwidth requests from child nodes (incoming traffic).

Step 2: Collect the node bandwidth requirement from the application (self/local traffic).

Step 3: Collect the current outgoing scheduled bandwidth (outgoing traffic).

Step 4: If (outgoing < incoming + self) then SCHEDULE soft cells/ bundles to satisfy bandwidth requirements.

Step 5: If (outgoing > incoming + self) then DELETE the soft cells that are not used.

Step 6: Return to step 1.

The default bandwidth estimation algorithm introduced in this document adopts a reactive allocation policy; it is possible to configure proactivity by using a given PROACTIVETHRESH value. In this case, at Step 4, new soft cells will be scheduled, using the cell allocation method, only if there are no free cells in the bundle that can satisfied the current bandwidth request. The node asks 6top for increasing the bundle size by using the bundle allocation method.

8. OTF external CoAP interface

In order to select the current OTF algorithm and provide functional parameters from outside OTF, this module uses CoAP with YANG as the data model. The algorithm number and the parameters MUST be invoked in different CoAP calls.

The path to select the algorithm is '6t/e/otf/alg' with A as the algorithm number.

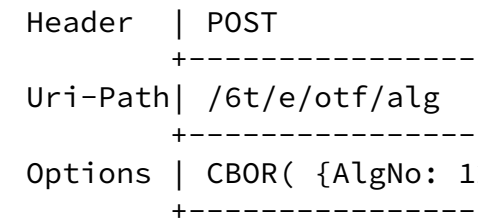

Figure 1: Algorithm number POST message

To obtain the current algorithm number:

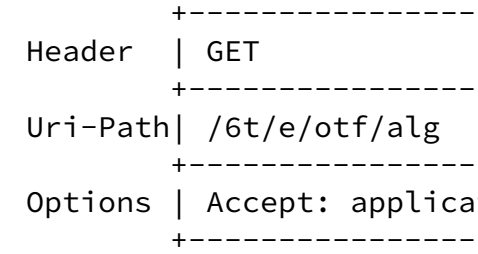

Figure 2: Algorithm number GET message

An example is: 'coap://[aaaa::1]/6t/e/otf/alg'

The current algorithm parameter path is '6t/e/otf/alg/par'.

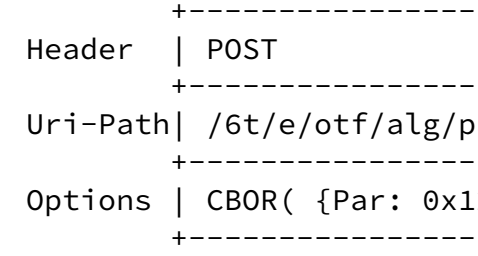

Figure 3: Algorithm number POST message

An example follows: 'coap://[aaaa::1]/6t/e/otf/alg/par'

9. Acknowledgments

Special thanks to Prof. Kris Pister for his valuable contribution in designing the default Bandwidth Estimation Algorithm, and to Prof.

Qin Wang for her support in defining the interaction between OTF and 6top sublayer.

Thanks to the Fondecyt 1121475 Project, to INRIA Chile "Network Design" group and to the IoT6 European Project (STREP) of the 7th Framework Program (Grant 288445).

10. References

10.1. Informative References

[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-01 (work in progress), February 2014.

[I-D.ietf-6tisch-architecture]
Thubert, P., Watteyne, T., and R. Assimiti, "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-architecture-02 (work in progress), June 2014.

[I-D.ietf-6tisch-tsch]
Watteyne, T., Palattella, M., and L. Grieco, "Using IEEE802.15.4e TSCH in an LLN context: Overview, Problem Statement and Goals", draft-ietf-6tisch-tsch-00 (work in progress), November 2013.

[I-D.wang-6tisch-6top]
Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top)", draft-wang-6tisch-6top-00 (work in progress), October 2013.

10.2. External Informative References

[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.

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