6TiSCH
Internet-Draft
Intended status: Informational
Expires: September 8, 2015

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

March 7, 2015

6TiSCH On-the-Fly Scheduling

draft-dujovne-6tisch-on-the-fly-05

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 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.

Status of This Memo

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 algorithm 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 soft cells. It

also defines two threshold values for bounding the number of triggered 6top allocate/delete commands. 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 soft cells) 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, OTF 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 interchangeably.

OTF is a Layer-3 Mechanism, and as such, it operates on L3 links, on the best effort track, i.e. with TrackID=00, as defined in [I-D.wang-6tisch-6top]. Inside an intermediate node, a track is uniquely associated to a pair of bundles: one incoming bundle, and one outgoing bundle. For an IP link, the two bundle are identified by the same peer mac addresses. For instance (macA, macB, TrackID=00) and (macB, macA, TrackID=00) will be the two bundles associated to the L3 link between node A and node B. The cells on the best effort track can be used for forwarding any packet in the queue, regardless of the specific L2 bundle (and thus, end-to-end L2 track) the packet 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.

This document introduces the following parameters for describing the behavior of the OTF allocation policy:

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

REQUIREDCELLS: Number of cells requested by OTF to 6top, a non-negative value. How this is computed is out of the scope. It MAY be an instantaneous request, or a value averaged on several measurements.

OTFTHRESHLOW: Threshold parameter introducing cell over-provisioning in the allocation policy. It is a non-negative value expressed as number of cells. Which value to use is application-specific and out of scope.

OTFTHRESHHIGH: Threshold parameter introducing cell under-provisioning in the allocation policy. It is a non-negative value expressed as number of cells. Which value to use is application-specific and out of scope.

The OTF allocation policy compares the number of required cells against the number of scheduled ones, using the OTF threshold for bounding the signaling overhead due to negotiations of new cells. In details:

```
                SCHEDULEDCELLS
    <-------------------------------------->
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+
   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+
                       |<----------------->|<------------->|
                       |   OTFTHRESHLOW    | OTFTHRESHHIGH |
                       |                   |               |
   REQUIREDCELLS       |                   |               |
   +---+---+---+       |                   |               |     ADD
   |   |   |   |       |                   |               |     SOME
   +---+---+---+       |                   |               |     CELLS
                       |                   |               |
           REQUIREDCELLS                   |               |
   +---+---+---+---+---+---+---+           |               |     DO
   |   |   |   |   |   |   |   |           |               |     NOTHING
   +---+---+---+---+---+---+---+           |               |
                       |                   |               |
                   REQUIREDCELLS           |               |
   +---+---+---+---+---+---+---+---+---+---+---+           |     DO
   |   |   |   |   |   |   |   |   |   |   |   |           |     NOTHING
   +---+---+---+---+---+---+---+---+---+---+---+           |
                       |                   |               |
                       |   REQUIREDCELLS   |               |
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+ REMOVE
   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | SOME
   +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+ CELLS
```

Figure 1: Relation among the OTF parameters used for triggering add/ remove 6top commands

1. If REQUIREDCELLS is greater than (SCHEDULEDCELLS + OTFTHRESHHIGH), OTF asks 6top to add one or more soft cells to the bundle on the best effort track.

2. If REQUIREDCELLS is greater or equal than (SCHEDULEDCELLS - OTFTHRESHLOW), and it is lower than or equal to (SCHEDULEDCELLS + OTFTHRESHHIGH), OTF does not perform any bundle resizing, since the scheduled cells are sufficient for managing the current traffic conditions.

3. If REQUIREDCELLS is lower than (SCHEDULEDCELLS - OTFTHRESHLOW), OTF asks 6top to delete one or more soft cells from the bundle on the best-effort track.

When both OTFTHRESHLOW and OTFTHRESHHIGH equal 0, any discrepancy between REQUIREDCELLS and SCHEDULEDCELLS triggers a 6top negotiation

of soft cells. Other values for the thresholds values reduce the number of triggered 6top negotiations.

The number of soft cells to be scheduled/deleted for bundle resizing is out of the scope of this document and implementation-dependant.

3. Allocation method

Beyond the allocation policies that describe the approach used by OTF for fulfilling the node bandwidth requests, the OTF framework also includes the Allocation Method that specify how OTF issues commands to the 6top sublayer. As specified in [I-D.wang-6tisch-6top], 6top provides a set of commands that allows OTF to allocate/delete soft cells. Such commands are used by the OTF soft cell allocation method.

With the soft cell allocation method, OTF can ask 6top to reserve one (or N > 1) soft cell(s) on the best effort L3 boundle, between two neighbor nodes. The 6top layer allocates and maintains these cells. If a L3 bundle with TrackID=00 was already reserved between the same pair of neighbors, 6top translates the OTF request into a bundle resize request. The newly allocated cell increases 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 (or N > 1) from the best effort track, i.e. to decrease the size of the best effort L3 bundle. If no bundle with TrackID=00 exists when 6top receives the OTF request, then the 6top softcell create command generates a new bundle of size 1.

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 specifies the track the cell belongs to (i.e., best effort track, TrackID=00), but not its slotOffset nor the channelOffset. If at least one cell on the best effort L3 bundle 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. By using the CREATE.softcell command, OTF can ask 6top to add multiple softcells on the best effort L3 bundle. 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 L3 bundle size of the already existing best-effort

bundle. By using the DELETE.softcell command, OTF can ask 6top to delete cells from the 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 structures 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 drops 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, e.g. based on the queue length increase, OTF may ask 6top to add some cells, in order to increase the available bandwidth.

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 (DELETE.softcell) 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 soft cell allocation. Both type of events, the one which triggers the algorithm and the ones which are generated by the execution of the algorithm are called OTF events.

A set of parameters P(E): parameters used to define E and its triggering conditions;

a set of triggering variables V(E): variables that can trigger the event;

a set of triggering conditions C(E): conditions to satisfy on the variables V(E) to trigger E;

a set of process handlers H(E): handlers required to respond and process the triggering conditions C(E).

To illustrate how P(E), V(E), C(E) and H(E) can be used to define a real event, the allocation policy described in Sec. 2 is considered hereby.

P(E) consists of the OTFTHRESHLOW and OTFTHRESHHIGH parameters (P1 and P2, respectively);

V(E) consists of the REQUIREDCELLS and SCHEDULEDCELLS parameters (V1 and V2, respectively);

C(E) consists of the following conditions:

C1: V1 > V2+P2

C2: V1 <=V2-P1

H(E) consists of the following handlers (one handler for each triggering condition)

H1(C1): OTF asks 6top to add one or more soft cells to the L3 best effort bundle.

H2(C2): OTF asks 6top to delete one or more soft cells from the L3 best effort bundle.

7. Bandwidth Estimation Algorithms

OTF supports different bandwidth estimation algorithms that can be used by a node in a 6TiSCH network for checking the statistics provided by 6top and the actual bandwidth usage. By doing so, one can adapt (increase or decrease) the number of scheduled soft cells for a given pair of neighbors (e.g., parent node and its child), according to their specific requirements. 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.

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 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, i.e., it uses OTFTHRESHLOW = 0 and OTFTHRESHHIGH = 0.

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.

```
          +-------------------------------------------+
 Header   | POST                                      |
          +-------------------------------------------+
 Uri-Path| /6t/e/otf/alg                              |
          +-------------------------------------------+
 Options | CBOR( {AlgNo: 123} )                       |
          +-------------------------------------------+
```

Figure 2: Algorithm number POST message

To obtain the current algorithm number:

```
          +-------------------------------------------+
 Header   | GET                                       |
          +-------------------------------------------+
 Uri-Path| /6t/e/otf/alg                              |
          +-------------------------------------------+
 Options | Accept: application/cbor                   |
          +-------------------------------------------+
```

Figure 3: 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'.

```
          +-------------------------------------------+
 Header   | POST                                      |
          +-------------------------------------------+
 Uri-Path| /6t/e/otf/alg/par                          |
          +-------------------------------------------+
 Options | CBOR( {Par: 0x1234} )                      |
          +-------------------------------------------+
```

Figure 4: 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-03 (work in progress), January 2015.

[I-D.ietf-6tisch-architecture]
Thubert, P., Watteyne, T., Struik, R., and M. Richardson, "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-architecture-05 (work in progress), January 2015.

[I-D.ietf-6tisch-tsch]
Watteyne, T., Palattella, M., and L. Grieco, "Using IEEE802.15.4e TSCH in an IoT context: Overview, Problem Statement and Goals", draft-ietf-6tisch-tsch-05 (work in progress), January 2015.

[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