

6TiSCH
Internet-Draft
Intended status: Informational
Expires: April 12, 2014

X. Vilajosana, Ed.
Universitat Oberta de Catalunya
K. Pister
University of California Berkeley
October 09, 2013

Minimal 6TiSCH Configuration
draft-vilajosana-6tisch-minimal-00

Abstract

This document describes the minimal set of rules to operate a [[IEEE802154e](#)] Timeslotted Channel Hopping (TSCH) network. This minimal mode of operation can be used during network bootstrap, as a fallback mode of operation when no dynamic scheduling solution is available or functioning, or during early interoperability testing and development.

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 April 12, 2014.

Copyright Notice

Copyright (c) 2013 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

Internet-Draft

6tisch-minimal

October 2013

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Minimal Schedule Configuration	3
2.1.	Slotframe	3
2.2.	Cell Options	5
2.3.	Retransmissions	6
2.4.	Time Slot timing	6
3.	Enhanced Beacons Configuration and Content	7
3.1.	Sync IE	8
3.1.1.	IE Header	8
3.1.2.	IE Content	8
3.2.	Frame and Cell IE	8
3.2.1.	IE Header	9
3.2.2.	IE Content	9
4.	Acknowledgement	9
4.1.	ACK/NACK Time Correction IE	9
4.1.1.	IE Header	9
4.1.2.	IE Content	10
5.	Neighbour information	10
5.1.	Neighbour Table	10
5.2.	Time Source Neighbour Selection	11
6.	Queues and Priorities	11
7.	RPL on TSCH	12
7.1.	RPL Objective Function Zero	12
7.1.1.	Rank computation	12
7.1.2.	Rank computation Example	13
7.2.	RPL Configuration	15
7.2.1.	Mode of Operation	15
7.2.2.	Trickle Timer	16
7.2.3.	Hysteresis	16
8.	Acknowledgements	16
9.	References	16
9.1.	Normative References	16
9.2.	Informative References	17
9.3.	External Informative References	18
	Authors' Addresses	18

[1.](#) Introduction

The nodes in a [[IEEE802154e](#)] TSCH network follow a communication schedule. The entity (centralized or decentralized) responsible for building and maintaining that schedule has very precise control over the trade-off between the network's latency, bandwidth, reliability and power consumption. During early interoperability testing and

development, however, simplicity is often more important than efficiency. One goal of this document is to define the simplest set of rules for building a [[IEEE802154e](#)] TSCH-compliant network, at the necessary price of lesser efficiency. Yet, this minimal mode of operation can also be used during network bootstrap before any schedule is installed into the network so nodes can self organize and the management and configuration information be distributed. In addition, as outlined in [[I-D.phinney-roll-rpl-industrial-applicability](#)] the minimal configuration can be used as a fallback mode of operation, ensuring connectivity of nodes in case that dynamic scheduling mechanisms fail or are not available. [[IEEE802154e](#)] provides a mechanism whereby the details of slotframe length, timeslot timing, and channel hopping pattern are communicated at synchronization to a node, also Enhanced Beacons can be used to periodically update nodes information. This document describes specific settings for these parameters. Nodes SHOULD broadcast properly formed Enhanced Beacons to announce these values, but during initial implementation and debugging it may be convenient to hard-code these values.

[2.](#) Minimal Schedule Configuration

In order to form a network, a minimum schedule configuration is required so nodes can advertise the presence of the network, and allow other nodes to join.

[2.1.](#) Slotframe

The slotframe, as defined in [[I-D.palattella-6tsch-terminology](#)], is an abstraction of the MAC layer that defines a collection of time slots of equal length and priority, and which repeats over time. In order to set up a minimal TSCH network, nodes need to be synchronized with the same slotframe configuration so they can exchange Enhanced Beacons (EBs) and data packets. This document recommends the following slotframe configuration.

Property	Value
Number of time slots per Slotframe	101
Number of available channels	16
Number of EBs cells	1 (slotOffset 0)
Number of scheduled cells	5 (slotOffsets 1,2,3,4,5)
Number of unscheduled cells	95 (from slotOffset 6 to 100)
Number of MAC retransmissions (max)	3
Time Slot duration	15ms

The suggested minimal schedule may be hard-coded in each node. The slotframe is composed of 101 time slots. The first slot in the slotframe is used to send Enhanced Beacons announcing the presence of the network. These EBs are not acknowledged. Five cells are scheduled for exchanging data packets, as described in [Section 2.2](#). These cells are scheduled at slotOffset 1 to 5, and channelOffset 0. Per the IEEE802.15.4e TSCH, data packets sent on these cells to a unicast MAC address are acknowledged by the receiver. The 95

Minimal schedule overview

- EB: Enhanced Beacon
- Tx: Transmit
- Rx: Receive
- S: Shared
- OFF: Unscheduled (can be used by a dynamic scheduling mechanism)

Per the [IEEE802154e] TSCH, each scheduled cell has a bitmap of cell

options assigned, named LinkOption. All scheduled cells in the minimal schedule are configured as Hard cells [\[I-D.watteyne-6tsch-tsch-lln-context\]](#) [\[I-D.wang-6tsch-6top\]](#). Additional available cells can be scheduled by a dynamic scheduling solution and can either be configured as hard cells or soft cells without any restriction.

The EB cell is assigned the following bitmap of cell options:

b0 = Transmit = 1 (set)

b1 = Receive = 0 (clear)

b2 = Shared = 0 (clear)

b3 = Timekeeping = 0 (clear)

b4 = Hard = 1 (set)

b5-b7 = Reserved (clear)

The data cells are assigned the bitmap of cell options below that results in "Slotted Aloha" behaviour. Because both the "Transmit" and "Receive" bits are set, a node either transmits, if there is a packet in its queue, or listens if it has nothing to transmit. Because the "shared" bit is set, the back-off mechanism defined in [\[IEEE802154e\]](#) is used to resolve contention.

b0 = Transmit = 1 (set)

b1 = Receive = 1 (set)

b2 = Shared = 1 (set)

b3 = Timekeeping = 0 (clear)

b4 = Hard = 1 (set)

b5-b7 = Reserved (clear)

All remaining cells are unscheduled. Thus the nodes can keep their radio off. In a memory efficient implementation, scheduled cells could be represented by a circular linked list. Unscheduled cells SHOULD NOT occupy any memory.

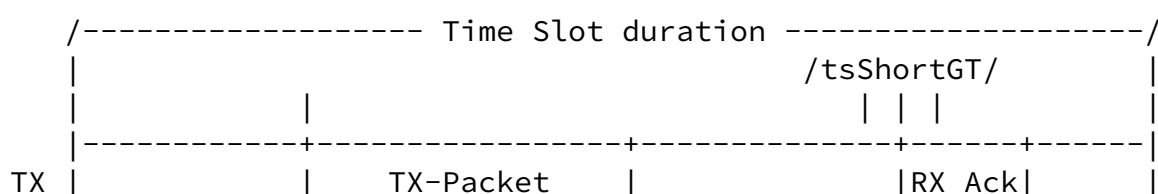
[2.3.](#) Retransmissions

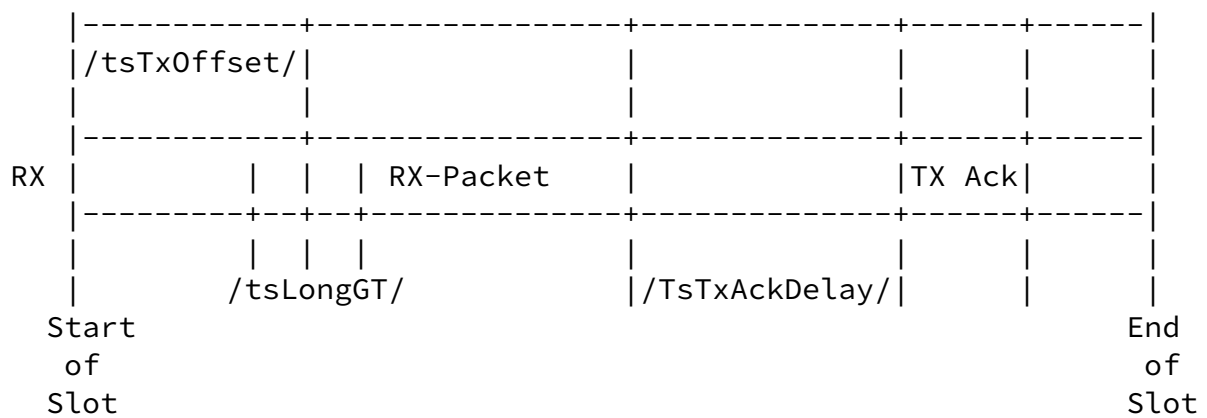
The maximum number of MAC-layer retransmissions is set to 3. For packets which require an acknowledgement, if none is received after a total of 4 attempts, the transmissions is considered failed and the MAC layer MUST notify the upper layer. Packets sent to the broadcast MAC address (including EBs) are not acknowledged and therefore not retransmitted.

[2.4.](#) Time Slot timing

The figure below shows an active timeslot in which a packet is sent from the transmitter node (TX) to the receiver node (RX). A MAC acknowledgement is sent back from the RX to the TX node, indicating successful reception. The TsTxOffset duration defines the instant in the timeslot when the first byte of the transmitted packet leaves the radio of the TX node. The radio of the RX node is turned on TsLongGT/2 before that instant, and listen for at least TsLongGT. This allows for a de-synchronization between the two node of at most TsLongGT. The RX node needs to send the first byte of the MAC acknowledgement exactly TsTxAckDelay after the end of the last byte of the received packet. TX's radio has to be turned on TsShortGT/2 before that time, and keep listening for at least TsShortGT.

Time slot internal timing diagram





[IEEE802154e] does not define the different durations of a time slot. It does allow those durations to be sent in the EBs (through a TimeSlot IE). This document recommends to pre-configure the different durations to the values listed below or use EBs to learn those values included in the TimeSlot IE.

Timeslot durations

IEEE802.15.4e TSCH parameter	Value
TsTxOffset	4000us
TsLongGT	2600us
TsTxAckDelay	4606us
TsShortGT	1000us
Time Slot duration	15000us

3. Enhanced Beacons Configuration and Content

[IEEE802154e] does not define how often or which EBs are sent. The choice of the duration between two EBs needs to take into account

whether EBs are used as the only mechanism to synchronize devices, or

whether a Keep-Alive (KA) mechanism is used in parallel. For a simplest TSCH configuration, a mote SHOULD send an EB every 10s. For additional reference see [[I-D.watteyne-6tsch-tsch-lln-context](#)] where different synchronization approaches are summarized.

EBs MUST be sent with the Beacon IEEE802.15.4 frame type and this EBs MUST carry the following Information Elements (IEs): (The content of the IEs is presented here for clarity, however this information is redundant with [[I-D.watteyne-6tsch-tsch-lln-context](#)] and [[IEEE802154e](#)].)

[3.1.](#) Sync IE

Contains synchronization information such as ASN and Join Priority. The value of Join Priority is discussed in [Section 5.2](#).

[3.1.1.](#) IE Header

Length (b0-b7) = 0x06

Sub-ID (b8-b14) = 0x1a

Type (b15) = 0x00 (short)

[3.1.2.](#) IE Content

ASN Byte 1 (b16-b23)

ASN Byte 2 (b24-b31)

ASN Byte 3 (b32-b39)

ASN Byte 4 (b40-b47)

ASN Byte 5 (b48-b55)

Join Priority (b56-b63)

[3.2.](#) Frame and Cell IE

Although the schedule may be hard-coded during development, each node MUST indicate the schedule in each EB through a Frame and Cell IE. This enables nodes which implement [[IEEE802154e](#)] fully to configure their schedule as they join the network, and interact with nodes using a hard-coded schedule.

[3.2.1.](#) IE Header

Length (b0-b7) = variable

Sub-ID (b8-b14) = 0x1b

Type (b15) = 0x00 (short)

[3.2.2.](#) IE Content

Slotframes (b16-b23) = 0x01

Slotframe ID (b24-b31) = 0x01

Size Slotframe (b32-b47) = 0x65 (101)

Links (b48-b55) = 0x06

For each link in the minimal schedule:

Channel Offset (2B) = 0x00

Slot Number (2B) = from 0x00 to 0x05

LinkOption (1B) = as described in [Section 2.2](#)

[4.](#) Acknowledgement

MAC-layer acknowledgement frames are built according to [\[IEEE802154e\]](#). Data frames and command frames sent to a unicast MAC destination address request an acknowledgement. The acknowledgement frame is of type ACK (0x10). Each acknowledgement contains the following IE:

[4.1.](#) ACK/NACK Time Correction IE

The ACK/NACK time correction IE is used to carry the measured de-synchronization between the sender and the receiver.

[4.1.1.](#) IE Header

Length (b0-b7) = 0x02

Sub-ID (b8-b14) = 0x1e

Type (b15) = 0x00 (short)

Internet-Draft

6tisch-minimal

October 2013

[4.1.2.](#) IE Content

Time Synch Info and ACK status (b16-b31)

The possible values for the Time Synch Info and ACK status are described in [[IEEE802154e](#)] and reproduced in the following table:

ACK status and Time Synchronization information.

ACK Status	Value
ACK with positive time correction	0x0000 - 0x07ff
ACK with negative time correction	0x0800 - 0x0fff
NACK with positive time correction	0x8000 - 0x87ff
NACK with negative time correction	0x8800 - 0x8fff

[5.](#) Neighbour information

[[IEEE802154e](#)] does not define how and when each node in the network keeps information about its neighbours. This document recommends to keep the following information in the Neighbour table:

[5.1.](#) Neighbour Table

The exact format of the neighbour table is implementation-specific, but it SHOULD contain the following information for each neighbour:

Neighbour statistics:

numTx: number of transmitted packets to that neighbour

numTxAck: number of transmitted packets that have been acknowledged by that neighbour

numRx: number of received packets from that neighbour

The EUI64 of the neighbour address.

Timestamp when that neighbour was heard for the last time. This can be based on the ASN counter or any other time base. Can be used to trigger a keep-alive message.

RPL rank of that neighbour.

A flag which indicates whether this neighbour is a time source neighbour.

Connectivity statistics (e.g., RSSI), which can be used to determine the quality of the link.

In addition to that information, each node has to be able to compute some RPL Objective Function (OF) taking into account the neighbour and connectivity statistics. An example RPL objective function is the OF Zero as described in [\[RFC6552\]](#) and [Section 7.1.1](#).

[5.2](#). Time Source Neighbour Selection

Each node MUST select at least one time source neighbour amongst its known neighbours in its RPL routing parent set. When a node joins a network, it has no routing information yet. To select its time source neighbour, uses the Join Priority information advertised in the EB as described in [Section 5.2.4.13](#) and Table 52b of [\[IEEE802154e\]](#). The Sync IE contains the ASN and 1 Byte field named Join Priority. The Join Priority of any node is equivalent to the result of the function DAGRank(rank) as defined by [\[RFC6550\]](#) and [Section 7.1.1](#). The Join Priority of the DAG root is zero, i.e., EBs sent from the DAG root are sent with Join Priority equal to 0. A lower value of the Join Priority indicates that the device is the preferred one to connect to. When a node Joins the network MUST NOT be allowed to send EBs until it has acquired a RPL rank. The latter avoids topology loops and matches RPL topology with underlying mesh topology. As soon as a node acquires a RPL rank (see [\[RFC6550\]](#) and [Section 7.1.1](#)), it SHOULD send Enhanced Beacons including a Sync IE with Join Priority field set as DAGRank(rank) where rank is the rank

of the actual node. In case of a node receives EBs from different nodes with equal Join Priority, the time source neighbour selection should be assessed by other metrics that can help to determine the better connectivity link. Time source neighbor hysteresis SHOULD be addressed according to the rules defined in [Section 7.2.3](#). If connectivity to the time source neighbor is lost, a new time source neighbor MUST be chosen among the neighbor in the RPL routing parent set.

Optionally, some form of hysteresis SHOULD be implemented to avoid frequent changes in time source neighbors.

[6.](#) Queues and Priorities

[IEEE802154e] does not define the use of queues to handle upper layer data (either application or control data from upper layers). This

document recommends the use of a single queue with the following rules:

When the node is not synchronized to the network, higher layers are not able to insert packets into the queue.

Frames generated by the MAC layer (e.g., EBs and ACK) have a higher priority than packets received from a higher layer.

IEEE802.15.4 frames of types Beacon and Command have a higher priority than IEEE802.15.4 frames of types Data and ACK.

One entry in the queue is reserved at all times for an IEEE802.15.4 frames of types Beacon or Command frames.

[7.](#) RPL on TSCH

Nodes in the network MUST use the RPL routing protocol

[7.1.](#) RPL Objective Function Zero

Nodes in the network MUST use the RPL routing protocol [[RFC6550](#)].

[7.1.1.](#) Rank computation

The rank computation is described at [\[RFC6552\] Section 4.1](#). Briefly, a node rank is computed by the following equation:

$$R(N) = R(P) + \text{rank_increase}$$

$$\text{rank_increase} = (R_f * S_p + S_r) * \text{MinHopRankIncrease}$$

Where:

$R(N)$: Rank of the node.

$R(P)$: Rank of the parent obtained as part of the DIO information.

rank_increase : The result of a function that determines the rank increment.

R_f (rank_factor): A configurable factor that is used to multiply the effect of the link properties in the rank_increase computation. If none is configured, then a rank_factor of 1 is used. For the purpose of this document rank_factor MUST be set to 1.

S_p (step_of_rank): (strictly positive integer) - an intermediate computation based on the link properties with a certain neighbour. For the purpose of this document $2 * \text{ETX}$ (Expected Transmissions) as defined by [\[DeCouto03\]](#) and [\[RFC6551\]](#) MUST be used. The ETX will be computed as the inverse of the Packet Delivery Ratio (PDR) computed as the number of acknowledged packets divided by the number of transmitted packets to a certain node. E.g: $S_p = 2 * \text{numTX} / \text{numTXAck}$

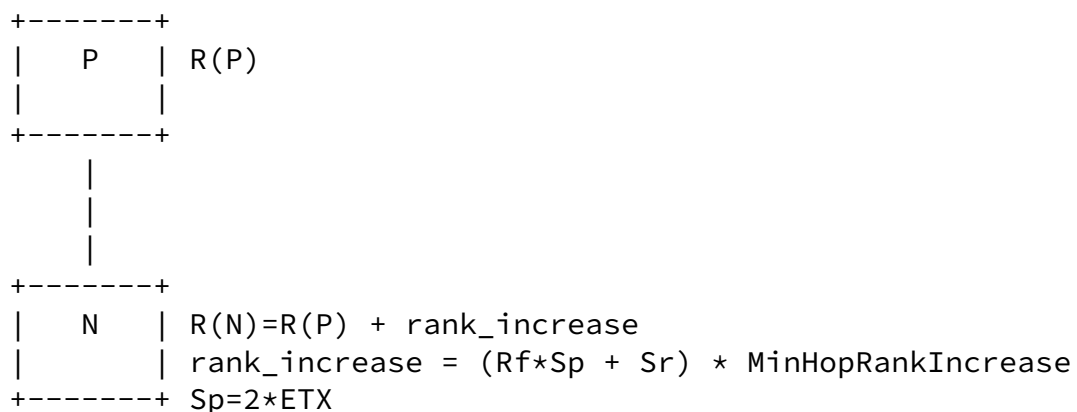
S_r (stretch_of_rank): (unsigned integer) - the maximum augmentation to the step_of_rank of a preferred parent to allow the selection of an additional feasible successor. If none is configured to the device, then the step_of_rank is not stretched. For the present document stretch_of_rank MUST be set to 0.

$\text{MinHopRankIncrease}$: the $\text{MinHopRankIncrease}$ is set to the fixed constant $\text{DEFAULT_MIN_HOP_RANK_INCREASE}$ [\[RFC6550\]](#). $\text{DEFAULT_MIN_HOP_RANK_INCREASE}$ has a value of 256.

DAGRank(rank): Equivalent to the floor of $(R_f * S_p + S_r)$ as defined by [RFC6550]. Specifically, when an Objective Function computes Rank this is defined as an unsigned integer (i.e., 16-bit) Rank quantity. When the Rank is compared, e.g., for determination of parent relationships or loop detection, the integer portion of the Rank is used. The integer portion of the Rank is computed by the DAGRank() macro as $\text{floor}(x)$ where $\text{floor}(x)$ is the function that evaluates to the greatest integer less than or equal to x .

$\text{DAGRank}(\text{rank}) = \text{floor}(\text{rank} / \text{MinHopRankIncrease})$

Rank computation scenario



[7.1.2.](#) Rank computation Example

This sections illustrates with an example the use of the Objective Function Zero. Assume the following parameters:

$R_f = 1$

$S_p = 2 * ETX$

$S_r = 0$

$\text{minHopRankIncrease} = 256$ (default in RPL)

$ETX = (\text{numTX} / \text{numTXAck})$

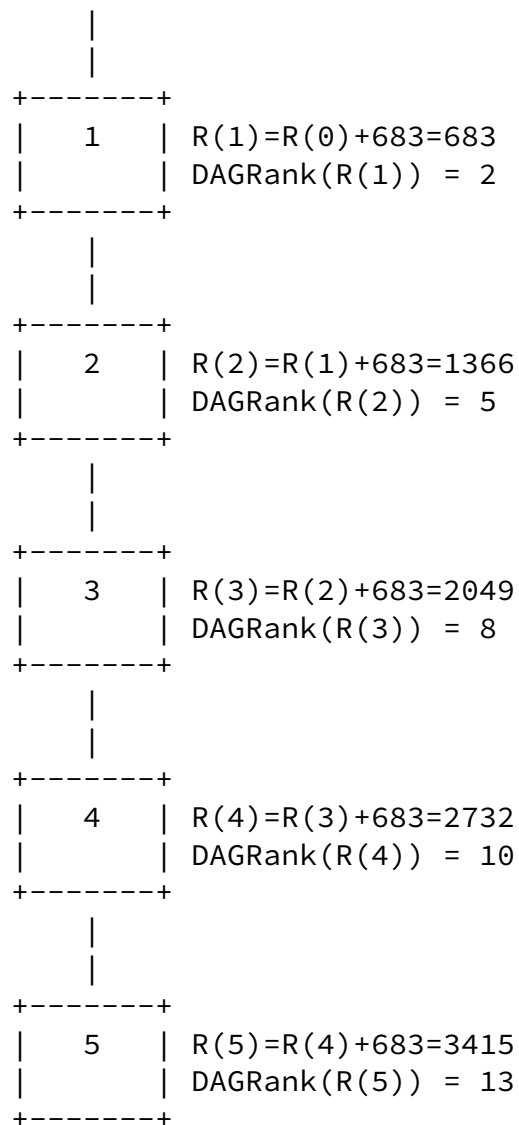
$r(n) = r(p) + \text{rank_increase}$

$\text{rank_increase} = (R_f * S_p + S_r) * \text{minHopRankIncrease}$

$\text{rank_increase} = 512 * \text{numTx} / \text{numTxACK}$

Rank computation example for 5 hop network where numTx=100 and numTxAck=75 for all nodes

```
+-----+
|   0   | R(0)=0
|       | DAGRank(R(0)) = 0
+-----+
```

[7.2.](#) RPL Configuration

In addition to the Objective Function (OF), a minimal configuration for RPL should indicate the preferred mode of operation and trickle timer operation so different RPL implementations can interoperate.

[7.2.1.](#) Mode of Operation

For downstream route maintenance, in a minimal configuration, RPL MUST be set to operate in the Non-Storing mode as described by [\[RFC6550\] Section 9.7](#). Storing mode ([\[RFC6550\] Section 9.8](#)) MAY be supported in less constrained devices.

[7.2.2.](#) Trickle Timer

RPL signalling messages such as DIOs are sent using the Trickle Algorithm [\[RFC6550\]](#) ([Section 8.3.1](#)) and [\[RFC6206\]](#). For the purpose of this document, the Trickle Timer MUST be used with the RPL defined default values [\[RFC6550\]](#) ([Section 8.3.1](#)). For a description of the Trickle timer operation see [Section 4.2](#) on [\[RFC6206\]](#).

[7.2.3.](#) Hysteresis

According to [\[RFC6552\]](#) the [\[RFC6719\]](#) recommends the use of a boundary value (PARENT_SWITCH_THRESHOLD) to avoid constant changes of parent when ranks are compared. When evaluating a parent that belongs to a smaller path cost than current minimum path, the candidate node is selected as new parent only if the difference between the new path and the current path is greater than the defined PARENT_SWITCH_THRESHOLD. Otherwise the node MAY continue to use the current preferred parent. As for [\[RFC6719\]](#) the recommended value for PARENT_SWITCH_THRESHOLD is 192 when ETX metric is used, the recommendation for this document is to use PARENT_SWITCH_THRESHOLD equal to 394 as the metric being used is 2*ETX. This mechanism is suited to deal with parent hysteresis in both cases routing parent and time source neighbor selection.

[8.](#) Acknowledgements

The authors would like to acknowledge the guidance and input provided by the 6TiSCH Chairs Pascal Thubert and Thomas Watteyne.

[9.](#) References

[9.1.](#) Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC6206] Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko, "The Trickle Algorithm", [RFC 6206](#), March 2011.
- [RFC6550] Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", [RFC 6550](#), March 2012.
- [RFC6551] Vasseur, JP., Kim, M., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks", [RFC 6551](#), March 2012.

Internet-Draft

6tisch-minimal

October 2013

- [RFC6552] Thubert, P., "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)", [RFC 6552](#), March 2012.
- [RFC6719] Gnawali, O. and P. Levis, "The Minimum Rank with Hysteresis Objective Function", [RFC 6719](#), September 2012.

[9.2](#). Informative References

- [I-D.watteyne-6tsch-tsch-lln-context]
Watteyne, T., Palattella, M., and L. Grieco, "Using IEEE802.15.4e TSCH in an LLN context: Overview, Problem Statement and Goals", [draft-watteyne-6tsch-tsch-lln-context-02](#) (work in progress), May 2013.
- [I-D.thubert-6tsch-architecture]
Thubert, P., Assimiti, R., and T. Watteyne, "An Architecture for IPv6 over Timeslotted Channel Hopping", [draft-thubert-6tsch-architecture-02](#) (work in progress), July 2013.
- [I-D.palattella-6tsch-terminology]
Palattella, M., Thubert, P., Watteyne, T., and Q. Wang, "Terminology in IPv6 over Timeslotted Channel Hopping", [draft-palattella-6tsch-terminology-01](#) (work in progress), July 2013.
- [I-D.wang-6tsch-6top]
Wang, Q., Vilajosana, X., and T. Watteyne, "6TSCH Operation Sublayer (6top)", [draft-wang-6tsch-6top-00](#) (work in progress), July 2013.
- [I-D.ohba-6tsch-security]
Chasko, S., Das, S., Lopez, R., Ohba, Y., Thubert, P., and A. Yegin, "Security Framework and Key Management Protocol Requirements for 6TSCH", [draft-ohba-6tsch-security-01](#) (work in progress), July 2013.
- [I-D.ietf-roll-terminology]
Vasseur, J., "Terms used in Routing for Low power And Lossy

Networks", [draft-ietf-roll-terminology-13](#) (work in progress), October 2013.

[I-D.phinney-roll-rpl-industrial-applicability]

Phinney, T., Thubert, P., and R. Assimiti, "RPL applicability in industrial networks", [draft-phinney-roll-rpl-industrial-applicability-02](#) (work in progress), February 2013.

Vilajosana & Pister

Expires April 12, 2014

[Page 17]

Internet-Draft

6tisch-minimal

October 2013

[9.3.](#) 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) Amendment 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.

[DeCouto03]

De Couto, D., Aguayo, D., Bicket, J., and R. Morris, "A High-Throughput Path Metric for Multi-Hop Wireless Routing", MobiCom '03, The 9th ACM International Conference on Mobile Computing and Networking, San Diego, California", June 2003.

[OpenWSN]

, "Berkeley's OpenWSN Project Homepage", ,
<<http://www.openwsn.org/>>.

Authors' Addresses

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

Phone: +34 (646) 633 681

Email: xvilajosana@uoc.edu

Kris Pister
University of California Berkeley
490 Cory Hall
Berkeley, California 94720
USA

Email: pister@eecs.berkeley.edu