6TiSCH X. Vilajosana, Ed.
Internet-Draft Universitat Oberta de Catalunya
Intended status: Informational K. Pister
Expires: July 10, 2015 University of California Berkeley
January 6, 2015

Minimal 6TiSCH Configuration
draft-ietf-6tisch-minimal-05

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 fall-back mode of operation when no dynamic scheduling solution is available or functioning, or during early interoperability testing and development.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction 3
2. Requirements Language 3
3. Minimal Schedule Configuration 3
3.1. Slotframe 3
3.2. Cell Options 5
3.3. Retransmissions 6
3.4. Time Slot timing 6
4. Enhanced Beacons Configuration and Content 8
4.1. Sync IE 9
4.1.1. IE Header 9
4.1.2. IE Content 9
4.2. TSCH Timeslot IE 9
4.2.1. IE Header 9
4.2.2. IE Content 10
4.3. Channel Hopping IE 10
4.3.1. IE Header 10
4.3.2. IE Content 10
4.4. Frame and Link IE 10
4.4.1. IE Header 10
4.4.2. IE Content 10
5. Acknowledgment 11
5.1. ACK/NACK Time Correction IE 11
5.1.1. IE Header 11
5.1.2. IE Content 11
6. Neighbor information 12
6.1. Neighbor Table 12
6.2. Time Source Neighbor Selection 13
7. Queues and Priorities 14
8. Security 14
9. RPL on TSCH 14
9.1. RPL Objective Function Zero 14
9.1.1. Rank computation 14
9.1.2. Rank computation Example 16
9.2. RPL Configuration 17
9.2.1. Mode of Operation 18
9.2.2. Trickle Timer 18
9.2.3. Hysteresis 18
9.2.4. Variable Values 18
10. Acknowledgments 19
11. References 19
11.1. Normative References 19
11.2. Informative References 20
11.3. External Informative References 21

Authors' Addresses 21

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 MAY 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 MAY be used as a fall-back 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. This document describes specific settings for these parameters. Nodes MUST broadcast properly formed Enhanced Beacons to announce these values.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. 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.

3.1. Slotframe

The slotframe, as defined in [I-D.ietf-6tisch-terminology], is an abstraction of the link layer that defines a collection of time slots of equal length, 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.

Minimal configuration

Property	Value
Number of time slots per Slotframe	Variable
Number of available frequencies	16
Number of scheduled cells	1 (slotOffset 0) (macLinkType NORMAL)
Number of unscheduled cells	The remainder of the slotframe
Number of MAC retransmissions (max)	3 (4 attempts to tx)

The slotframe is composed of a configurable number of time slots. Choosing the number of time slots per slotframe needs to take into account network requirements such as density, bandwidth per node, etc. In the minimal configuration, there is only a single active slot in slotframe, used to transmit data and EBs, and receive information. The trade-off between bandwidth, latency and energy consumption can be controlled by choosing a different slotframe length. The active slot MAY be scheduled at the slotOffset 0x00 and channelOffset 0x00 and MUST be announced in the EBs. EBs are sent using this active slot to the link-layer broadcast address (and are therefore not acknowledged). Data packets, as described in Section 3.2, use the same active slot. Per [IEEE802154e], data packets sent unicast on this cell are acknowledged by the receiver. The remaining cells in the slotframe are unscheduled, and MAY be used by dynamic scheduling solutions. Details about such dynamic scheduling solution are out of scope of this document.

The slotframe length (expressed in number of time slots) is configurable. The length used determines the duty cycle of the network. For example, a network with a 0.99% duty cycle is composed of a slotframe of 101 slots, which includes 1 active slot. The present document RECOMMENDS the use of a default slot duration set to 10ms and its corresponding default timeslot timings defined by the [IEEE802154e] macTimeslotTemplate. The use of the default macTimeslotTemplate MUST be announced in the EB by using the Timeslot IE containing only the default macTimeslotTemplateId. Other time slot durations MAY be supported and MUST be announced in the EBs. If one uses a timeslot duration different than 10ms, it is RECOMMENDED to use a power-of-two of 10ms (i.e. 20ms, 40ms, 80ms, etc.). In this case, EBs MUST contain the complete TimeSlot IE as described in

Section 3.4. This document also recommends to manufacturers to clearly indicate nodes not supporting the default timeslot value.

Example schedule with 0.99% duty cycle

```
   Chan.  +----------+----------+          +----------+
   Off.0  | TxRxS/EB |   OFF    |          |   OFF    |
   Chan.  +----------+----------+          +----------+
   Off.1  |          |          |   ...    |          |
          +----------+----------+          +----------+
              .
              .
              .
   Chan.  +----------+----------+          +----------+
   Off.15 |          |          |          |          |
          +----------+----------+          +----------+

slotOffset     0          1                    100
```

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

3.2. Cell Options

Per [IEEE802154e] TSCH, each scheduled cell has an associated bitmap of cell options, called LinkOptions. The scheduled cell in the minimal schedule is configured as a Hard cell [I-D.ietf-6tisch-tsch][I-D.ietf-6tisch-6top-interface]. Additional available cells MAY be scheduled by a dynamic scheduling solution. The dynamic scheduling solution is out of scope, and this specification does not make any restriction on the LinkOption associated with those dynamically scheduled cells (i.e. they can be hard cells or soft cells).

The active cell is assigned the bitmap of cell options below. Because both the "Transmit" and "Receive" bits are set, a node transmits if there is a packet in its queue, listens otherwise. Because the "shared" bit is set, the back-off mechanism defined in [IEEE802154e] is used to resolve contention when transmitting. This results in "Slotted Aloha" behavior. The "Timekeeping" flag is never set, since the time source neighbor is selected using the DODAG structure of the network (detailed below).

b0 = Transmit = 1 (set)

b1 = Receive = 1 (set)

b2 = Shared = 1 (set)

b3 = Timekeeping = 0 (clear)

b4-b7 = Reserved (clear)

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

3.3. Retransmissions

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

3.4. 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 link-layer acknowledgment is sent by the RX node to the TX node when the packet is to be acknowledged. 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 tsRxWait/2 before that instant, and listens for at least tsRxWait. This allows for a de-synchronization between the two nodes of at most tsRxWait. The RX node needs to send the first byte of the MAC acknowledgment exactly TsTxAckDelay after the end of the last byte of the received packet. TX's radio has to be turned on tsAckWait/2 before that time, and keep listening for at least tsAckWait. The TX node can perform a Clear Channel Assessment (CCA) if required, this does not interfere with the scope of this draft. As for a minimal configuration, CCA is not mandatory.

Time slot internal timing diagram

```
   /---------------------- Time Slot Duration ----------------------/
   |                                                  / (5) /       |
   |                   |              / tsRxAckDelay /|  |  |       |
   |-------------------+--------------+------------------+------+---|
TX |/(1)/  (2)  / (3) /|   TX packet  |                  |RX ack|   |
   |----+-------+------+--------------+------------------+------+---|
   |/    tsTxOffset   /|              |                  |      |   |
   |                   |              |                  |      |   |
   |-------------------+--------------+------------------+------+---|
RX |                |  |  | RX packet |                  |TX ack|   |
   |----------------+--+--+-----------+------------------+------+---|
   |                |  |  |           |                  |      |   |
   |               / (4) /           /   tsTxAckDelay   /      |   |
   Start                                                         End
   of                                                             of
   Slot                                                         Slot
/(1)/ tsCCAOffset
/(2)/ tsCCA
/(3)/ tsRxTx
/(4)/ tsRxWait
/(5)/ tsAckWait
```

A 10ms time slot length is the default value defined by [IEEE802154e]. Section 6.4.3.3.3 of [IEEE802154e] defines a default macTimeslotTemplate, i.e. the different duration within the slot. These values are summarized in the following table and MUST be used when utilizing the default time slot duration. In this case, the Timeslot IE only transports the macTimeslotTemplateId (0x00) as the timing values are well-known. If a timeslot template other than the default is used, the EB MUST contain a complete TimeSlot IE indicating the timeslot duration and the corresponding timeslot timings, requiring 25 bytes. Note however that in case of discrepancy between the values in this document and [IEEE802154e], the IEEE standard specification is leading.

Default timeslot durations (per [IEEE802154e], Section 6.4.3.3.3)

IEEE802.15.4e TSCH parameter	Value (us)
tsCCAOffset	1800
tsCCA	128
tsTxOffset	2120
tsRxOffset	1120
tsRxAckDelay	800
tsTxAckDelay	1000
tsRxWait	2200
tsAckWait	400
tsRxTx	192
tsMaxAck	2400
tsMaxTx	4256
Time Slot duration	10000

4. Enhanced Beacons Configuration and Content

[IEEE802154e] does not define how often EBs are sent, nor their contents. EBs should not in general be used for synchronization. Synchronization is achieved via acknowledgements to normal packet traffic, and keepalives. For a minimal TSCH configuration, a mote SHOULD send an EB every EB_PERIOD. For additional reference see [I-D.ietf-6tisch-tsch] where different synchronization approaches are summarized. EBs are only authenticated and payload is not encrypted. Refer to the 6TiSCH architecture document [I-D.ietf-6tisch-architecture] for further details on security aspects.

EBs MUST be sent with the Beacon IEEE802.15.4 frame type and this EBs MUST carry the Information Elements (IEs) listed below.

The content of the IEs is presented here for completeness, however this information is redundant with [I-D.ietf-6tisch-tsch] and [IEEE802154e].

4.1. Sync IE

Contains synchronization information such as ASN and Join Priority. The value of Join Priority is discussed in Section 6.2.

4.1.1. IE Header

Length (b0-b7) = 0x06

Sub-ID (b8-b14) = 0x1a

Type (b15) = 0x00 (short)

4.1.2. 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)

4.2. TSCH Timeslot IE

Contains the timeslot template identifier. This specification uses the default timeslot template as defined in [IEEE802154e], Section 5.2.4.15.

4.2.1. IE Header

Length (b0-b7) = 0x01

Sub-ID (b8-b14) = 0x1c

Type (b15) = 0x00 (short)

4.2.2. IE Content

Timeslot Template ID (b0-b7) = 0x00

4.3. Channel Hopping IE

Contains the channel hopping template identifier. This specification uses the default channel hopping template, as defined in [IEEE802154e], Section 5.2.4.16.

4.3.1. IE Header

Length (b0-b7) = 0x01

Sub-ID (b8-b14) = 0x1d

Type (b15) = 0x00 (short)

4.3.2. IE Content

Channel Hopping Template ID (b0-b7) = 0x00

The default sequence for the 2.4GHz OQPSK PHY is [5, 6, 12, 7, 15, 4, 14, 11, 8, 0, 1, 2, 13, 3, 9, 10] per section 5.1.1a of [IEEE802154e]. Note however that in case of discrepancy between the values in this document and [IEEE802154e], the IEEE standard specification is leading.

4.4. Frame and Link IE

Each node MUST indicate the schedule in each EB through a Frame and Link IE. This enables nodes which implement [IEEE802154e] to configure their schedule as they join the network.

4.4.1. IE Header

Length (b0-b7) = variable

Sub-ID (b8-b14) = 0x1b

Type (b15) = 0x00 (short)

4.4.2. IE Content

Slotframes (b16-b23) = 0x01

Slotframe ID (b24-b31) = 0x01

Size Slotframe (b32-b47) = variable

Links (b48-b55) = 0x01

For the active cell in the minimal schedule:

Channel Offset (2B) = 0x00

Slot Number (2B) = 0x00

LinkOption (1B) = as described in Section 3.2

5. Acknowledgment

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

5.1. ACK/NACK Time Correction IE

The ACK/NACK time correction IE carries the measured de-synchronization between the sender and the receiver.

5.1.1. IE Header

Length (b0-b7) = 0x02

Sub-ID (b8-b14) = 0x1e

Type (b15) = 0x00 (short)

5.1.2. IE Content

Time Synchronization Information and ACK status (b16-b31)

The possible values for the Time Synchronization Information 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

6. Neighbor information

[IEEE802154e] does not define how and when each node in the network keeps information about its neighbors. Keeping the following information in the neighbor table is RECOMMENDED:

6.1. Neighbor Table

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

Neighbor statistics:

numTx: number of transmitted packets to that neighbor

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

numRx: number of received packets from that neighbor

The EUI64 of the neighbor.

Timestamp when that neighbor 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 neighbor.

A flag indicating whether this neighbor is a time source neighbor.

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 neighbor and connectivity statistics. An example RPL objective function is the OF Zero as described in [RFC6552] and Section 9.1.1.

6.2. Time Source Neighbor Selection

Each node MUST select at least one Time Source Neighbor among the nodes in its RPL routing parent set. When a node joins a network, it has no routing information. To select its time source neighbor, it uses the Join Priority field 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 9.1.1. 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 higher preference to connect to that device. When a node joins the network, it MUST NOT send EBs before having acquired a RPL rank. This avoids routing loops and matches RPL topology with underlying mesh topology. As soon as a node acquires a RPL rank (see [RFC6550] and Section 9.1.1), it SHOULD send Enhanced Beacons including a Sync IE with Join Priority field set to DAGRank(rank), where rank is the node's rank. If a node receives EBs from different nodes with equal Join Priority, the time source neighbor selection SHOULD be assessed by other metrics that can help determine the better connectivity link. Time source neighbor hysteresis SHOULD be used, according to the rules defined in Section 9.2.3. If connectivity to the time source neighbor is lost, a new time source neighbor MUST be chosen among the neighbors in the RPL routing parent set.

The decision for a node to select one Time Source Neighbor when multiple EBs are received is open to implementers. For example, a node MAY wait until one EB from NUM_NEIGHBOURS_TO_WAIT neighbors have been received to select the best Time Source Neighbor. This condition MAY apply unless a second EB is not received after MAX_EB_DELAY seconds. This avoids initial hysteresis when selecting a first Time Source Neighbor. Discrepancies on ASN received from different EB sources MUST be solved by waiting for another EB once the time source neighbor to join has been selected.

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

7. Queues and Priorities

[IEEE802154e] does not define the use of queues to handle upper layer data (either application or control data from upper layers). The use of a single queue with the following rules is RECOMMENDED:

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 frame types Beacon and Command have a higher priority than IEEE802.15.4 frame 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.

8. Security

As this document refers to the interaction between Layer 3 and Layer 2 protocols, this interaction MUST be secured by L2 security mechanisms as defined by [IEEE802154e]. Key distribution is out of scope of this document, but examples include pre-configured keys at the nodes, shared keys amongst peers or well-known keys. Refer to the 6TiSCH architecture document [I-D.ietf-6tisch-architecture] for further details on security aspects.

9. RPL on TSCH

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

9.1. RPL Objective Function Zero

Nodes in the network MUST use the RPL routing protocol [RFC6550] and implement the RPL Objective Function Zero [RFC6552].

9.1.1. 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) + rank_increase

rank_increase = (Rf*Sp + Sr) * 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.

Rf (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, rank_factor of 1 is used. In this specification, a rank_factor of 1 MUST be used.

Sp (step_of_rank): (strictly positive integer) - an intermediate computation based on the link properties with a certain neighbor. In this specification, 2*ETX (Expected Transmissions) as defined by [decouti03high] and [RFC6551] MUST be used. The ETX is computed as the inverse of the Packet Delivery Ratio (PDR), and MAY be computed as the number of acknowledged packets, divided by the number of transmitted packets to a certain node. E.g: Sp=2*numTX/numTXAck

Sr (stretch_of_rank): (unsigned integer) - the maximum increment 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. In this specification, stretch_of_rank MUST be set to 0.

MinHopRankIncrease: the MinHopRankIncrease is set to the fixed constant DEFAULT_MIN_HOP_RANK_INCREASE [RFC6550]. DEFAULT_MIN_HOP_RANK_INCREASE has a value of 256.

DAGRank(rank): Equivalent to the floor of (Rf*Sp + Sr) as defined by [RFC6550]. Specifically, when an Objective Function computes Rank, this is defined as an unsigned integer (i.e., a 16-bit value) Rank quantity. When the Rank is compared, e.g. to determine 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 floor(x) where floor(x) is the function that evaluates to the greatest integer less than or equal to x. DAGRank(rank) = floor(rank/MinHopRankIncrease)

Rank computation scenario

```
    +-------+
    |   P   | R(P)
    |       |
    +-------+
        |
        |
    +-------+
    |   N   | R(N)=R(P) + rank_increase
    |       | rank_increase = (Rf*Sp + Sr) * MinHopRankIncrease
    +-------+ Sp=2*ETX
```

9.1.2. Rank computation Example

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

```
Rf = 1

Sp = 2* ETX

Sr = 0

minHopRankIncrease = 256 (default in RPL)

ETX=(numTX/numTXAck)

r(n) = r(p) + rank_increase

rank_increase = (Rf*Sp + Sr) * minHopRankIncrease

rank_increase = 512*numTx/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
    +-------+
        |
        |
    +-------+
    |   1   | R(1)=R(0)+683=683
    |       | DAGRank(R(1)) = 2
    +-------+
        |
        |
    +-------+
    |   2   | R(2)=R(1)+683=1366
    |       | DAGRank(R(2)) = 5
    +-------+
        |
        |
    +-------+
    |   3   | R(3)=R(2)+683=2049
    |       | DAGRank(R(3)) = 8
    +-------+
        |
        |
    +-------+
    |   4   | R(4)=R(3)+683=2732
    |       | DAGRank(R(4)) = 10
    +-------+
        |
        |
    +-------+
    |   5   | R(5)=R(4)+683=3415
    |       | DAGRank(R(5)) = 13
    +-------+
```

9.2. 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 inter-operate. RPL information and hop-by-hop extension headers MUST be compressed according to the specification described in [I-D.thubert-6lo-rpl-nhc]

9.2.1. Mode of Operation

For downstream route maintenance, in a minimal configuration, RPL SHOULD 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.

9.2.2. Trickle Timer

RPL signaling messages such as DIOs are sent using the Trickle Algorithm [RFC6550] (Section 8.3.1) and [RFC6206]. For this specification, 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].

9.2.3. Hysteresis

According to [RFC6552], [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 is mechanism is suited to deal with parent hysteresis in both cases routing parent and time source neighbor selection.

9.2.4. Variable Values

The following table presents the RECOMMENDED values for the RPL-related variables defined in the previous section.

Recommended variable values

Variable	Value
EB_PERIOD	10s
MAX_EB_DELAY	180
NUM_NEIGHBOURS_TO_WAIT	2
PARENT_SWITCH_THRESHOLD	394

10. Acknowledgments

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

11. References

11.1. Normative References

[RFC6719] Gnawali, O. and P. Levis, "The Minimum Rank with Hysteresis Objective Function", RFC 6719, September 2012.

[RFC6552] Thubert, P., "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6552, 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.

[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.

[RFC6206] Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko, "The Trickle Algorithm", RFC 6206, March 2011.

[RFC3610] Whiting, D., Housley, R., and N. Ferguson, "Counter with CBC-MAC (CCM)", RFC 3610, September 2003.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

11.2. Informative References

[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-04 (work in progress), December 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-04 (work in progress), October 2014.

[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-02 (work in progress), July 2014.

[I-D.ietf-6tisch-6top-interface]
Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top) Interface", draft-ietf-6tisch-6top-interface-02 (work in progress), October 2014.

[I-D.richardson-6tisch-security-architecture]
Richardson, M., "security architecture for 6top: requirements and structure", draft-richardson-6tisch-security-architecture-02 (work in progress), April 2014.

[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.

[I-D.thubert-6lo-rpl-nhc]
Thubert, P. and C. Bormann, "A compression mechanism for the RPL option", draft-thubert-6lo-rpl-nhc-02 (work in progress), October 2014.

11.3. 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.

[CCM] National Institute of Standards and Technology, "Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality. SP 800-38C", May 2004.

[CCM-Star]
Struik, R., "Formal Specification of the CCM* Mode of Operation, IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs).", September 2005.

[decouti03high]
De Couto, D., Aguayo, D., Bicket, J., and R. Morris, "A High-Throughput Path Metric for Multi-Hop Wireless Routing", ACM International Conference on Mobile Computing and Networking (MobiCom) , June 2003.

[OpenWSN] Watteyne, T., Vilajosana, X., Kerkez, B., Chraim, F., Weekly, K., Wang, Q., Glaser, S., and K. Pister, "OpenWSN: a Standards-Based Low-Power Wireless Development Environment", Transactions on Emerging Telecommunications Technologies , August 2012.

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