6TiSCH X. Vilajosana, Ed.
Internet-Draft Universitat Oberta de Catalunya
Intended status: Standards Track K. Pister
Expires: March 24, 2016 University of California Berkeley
September 21, 2015

Minimal 6TiSCH Configuration
draft-ietf-6tisch-minimal-12

Abstract

This document describes the minimal set of rules to operate an IEEE 802.15.4 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

The nodes in a IEEE 802.15.4 TSCH network follow a communication
schedule. The entity (centralized or decentralized) responsible for
building and maintaining that schedule has 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 more important than efficiency. One goal of this document is to define the simplest set of rules for building a 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, 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. [IEEE802154] provides a mechanism whereby the details of slotframe length, timeslot timing, and channel hopping pattern are communicated when a node synchronizes to the network. 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 communicate. 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 transmission attempts)

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/receive both EBs and data link-layer frames. 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 [IEEE802154], 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 other (dynamic) scheduling solutions. Details about such dynamic scheduling solution are out of scope of this document. Details about the usage of the non scheduled cells 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 [IEEE802154] 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, EBs MUST contain the complete TimeSlot IE as described in Section 3.4. This document also recommends 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 [IEEE802154] 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 [RFC7554][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 [IEEE802154] is used to resolve contention when transmitting. This results in "Slotted Aloha" behavior. The "Timekeeping" flag is set so nodes initialy joining the network can maintin synchronization to the advertising node using that slot. Other time source neighbors

are selected using the DODAG structure of the network (detailed below).

b0 = Transmit = 1 (set)

b1 = Receive = 1 (set)

b2 = Shared = 1 (set)

b3 = Timekeeping = 1 (set)

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 transmission 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 bit after the Start of Frame Delimiter (SFD) 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/2 in either direction (early or late). The RX node needs to send the first bit after the SFD 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 document. As for a minimal configuration, CCA is OPTIONAL.

Time slot internal timing diagram

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

The timing parameters for the default macTimeslotTemplate (macTimeslotTemplateId = 0) MUST be used when utilizing the default time slot duration. In this case, the Timeslot IE only transports the macTimeslotTemplateId with value 0x00. 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. Note however that in case of discrepancy between the values in this document and [IEEE802154], the IEEE standard has precedence.

4. IEEE.802.15.4 Specific Header Fields and Considerations

The IEEE802.15.4 header of BEACON, DATA, ACKNOWLEDGEMENT, MAC COMMAND frames MUST include the Sequence Number field, the Source Address field and the Destination Address field. Frame Version field MUST be set to 0b10 (Frame Version 2).

The PAN ID Compression bit in a BEACON frame MUST indicate that the Source PAN ID is "Not Present" and the Destination PAN ID is "Present". The source address field MUST be filled with an extended address (64 bit) and this be indicated in the corresponding Frame Control field. The Destination address field MUST be filled with a short address (16bit) with a value of 0xffff to represent the broadcast short address.

A Node aiming to join the network by receving a properly formed BEACON shall enter in a scan phase and shall store the value of macPANId and then set it to 0xffff for the duration of the scan in order to meet the filtering rules in [IEEE802154].

When using DATA, ACKNOWLEDGEMENT, MAC COMMAND frame types the source and destination address fields MUST be filled with an extended address (64 bit) and this be indicated in the corresponding Frame Control field. The Destination PAN ID MUST be present, the Source PAN ID MUST be elided. The PAN ID Compression field MUST indicate that the Destination PAN ID is "Present" and the Source PAN ID is "Not Present". With [IEEE802154-2012] and according to Table 2a, this is accomplished by setting the PAN ID Compression bit to 0b0.

When preparing the security header, the ASN MUST be written into the Nonce in MSB format (most significant byte first) as indicated in [IEEE802154].

5. Enhanced Beacons Configuration and Content

[IEEE802154] does not define how often EBs are sent, nor their contents. EBs MUST 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 [RFC7554] where different synchronization approaches are summarized. EBs are only authenticated and neither Payload IEs nor the frame payload are encrypted.

EBs MUST be sent as per [IEEE802154] and MUST carry the Information Elements (IEs) listed below. Refer to Section 11.1 for an example of the Information Elements Header Content.

Synchronization IE: Contains synchronization information such as ASN and Join Priority. The value of Join Priority is discussed in Section 7.2.

TSCH Timeslot IE: Contains the timeslot template identifier. This specification uses the default timeslot template as defined in [IEEE802154]. In the case that a non-default timeslot template is used, the IE Content MUST follow the specification as defined in [IEEE802154] . Refer to Section 11.1 for an illustrative example of non default timeslot template.

Channel Hopping IE: Contains the channel hopping template identifier. This specification uses the default channel hopping template, as defined in [IEEE802154]. 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] [IEEE802154]. Note however that in case of discrepancy between the values in this document and [IEEE802154], the IEEE standard specification has preference.

Frame and Link IE: Each node MUST indicate the schedule in each EB through a Frame and Link IE. This enables nodes which implement [IEEE802154] to learn the schedule used in the network as they join it. This draft defines the use of a single cell set at channel offset 0x00, slot offset 0x00 and with linkOption 0x0F (TX, RX, SHARED bits set).

6. Acknowledgment

Link-layer acknowledgment frames are built according to [IEEE802154]. Unicast frames sent to a unicast MAC destination address MUST request an acknowledgment. The sender node MUST set the ACK requested bit in the MAC header. The acknowledgment frame is of type ACK (frame type value 0b010). Each acknowledgment contains the following IE:

ACK/NACK Time Correction IE: The ACK/NACK time correction IE carries the measured de-synchronization between the sender and the receiver. Refer to Section 11.3 for an example of the Header IE content. The possible values for the Time Synchronization Information and ACK status are described in [IEEE802154].

7. Neighbor information

[IEEE802154] 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:

7.1. 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 of the last frame received from that neighbor. This can be based on the ASN counter or any other time base. It 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 10.1.1.

7.2. 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 [IEEE802154]. The Sync IE contains the ASN and 1 Byte field named Join Priority. The Join Priority of any node MUST be equivalent to the result of the function DAGRank(rank)-1. The Join Priority of the DAG root is also equivalent to DAGRank(rank)-1. According to Section 10.1.1 the DAGRank(rank(0)) = 1 and therefore the DAGRank(rank(0))-1 is 0 which is compliant with the requirement of Join Priority = 0 imposed by [IEEE802154]. 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 10.1.1), it SHOULD send Enhanced Beacons including a Sync IE with Join Priority field set to DAGRank(rank)-1, 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 10.2.3. At any time, a node MUST maintain connectivity to at least one time source neighbor. New time source neighbors 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 implementation-specific.

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.

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

8. Queues and Priorities

[IEEE802154] 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 queuing priority than packets received from a higher layer.

Frame types Beacon and Command have a higher queuing priority than frame types Data and ACK.

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

9. 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 [IEEE802154]. Two security mechanisms are considered, authentication and encryption, authentication applies to all packet content while encryption applies to header IEs and MAC payload. Key distribution is out of scope of this document, but examples include pre-configured keys at the nodes, shared keys among peers or well-known keys.

The present document assumes the existence of two cryptographic keys, which can be pre-configured. One of the keys (K1) is used to authenticate EBs. As defined in Section Section 5, EBs MUST be authenticated, with no payload encryption. This facilitates logical segregation of distinct networks. A second key (K2) is used to authenticate DATA, ACKNOWLEDGEMENT, MAC COMMAND frame types and respective header IEs, with payload encryption. Depending on security policy, these keys could be the same (i.e., K1=K2).

For early interoperability, K1 MAY be set to 36 54 69 53 43 48 20 6D 69 6E 69 6D 61 6C 31 35 ("6TiSCH minimal15").

10. RPL on TSCH

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

10.1. RPL Objective Function Zero

Nodes in the multihop network MUST implement the RPL Objective Function Zero [RFC6552].

10.1.1. Rank computation

The rank computation is described at [RFC6552], Section 4.1. A node rank is computed by the following equation:

R(N) = R(P) + rank_increment

rank_increment = (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_increment: 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_increment computation. If none is configured, rank_factor of 1 is used. In this specification, a rank_factor of 1 SHOULD be used.

Sp (step_of_rank): (strictly positive integer) - an intermediate computation based on the link properties with a certain neighbor, i.e., the Expected Transmission Count (ETX) which provides an average of number of packet transmissions between two nodes. ETX is defined in detail by [decouto03high] and [RFC6551]. The ETX is computed as the inverse of the Packet Delivery Ratio (PDR), this is the number of transmitted packets, divided by the number of acknowledged packets to a certain node (e.g., ETX = numTX/numTXAck). An implementation MUST follow OF0's normalization guidance as discussed in Section 1 and Section 4.1 of [RFC6552], maintaining Sp between MINIMUM_STEP_OF_RANK of 1 to indicate a great quality and MAXIMUM_STEP_OF_RANK of 9 to indicate a really

poor quality, with DEFAULT_STEP_OF_RANK indicating a normal, average quality. One RECOMMENDED way to achieve this in an interoperable fashion is to set Sp to (3*ETX)-2.

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 SHOULD 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 "rank" 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). Nodes compute its DAGRank(rank) using DAGRank(R(N)).

Rank computation scenario

```
 +-------+
 |   P   | R(P)
 |       |
 +-------+
     |
     |
 +-------+
 |   N   | R(N)=R(P) + rank_increment
 |       | rank_increment = (Rf*Sp + Sr) * MinHopRankIncrease
 +-------+ Sp= (3*ETX) - 2
```

10.1.2. Rank computation Example

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

Rf = 1

Sp = (3*ETX)-2

Sr = 0

```
      minHopRankIncrease = 256 (default in RPL)

      ETX=(numTX/numTXAck)

      r(n) = r(p) + rank_increment

      rank_increment = (Rf*Sp + Sr) * minHopRankIncrease

      rank_increment = ((3*numTx/numTxAck)-2)*minHopRankIncrease = 512
```

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

```
       +-------+
       |   0   | R(minHopRankIncrease) = 256
       |       | DAGRank(R(0)) = 1
       +-------+
           |
           |
       +-------+
       |   1   | R(1)=R(0) + 512 = 768
       |       | DAGRank(R(1)) = 3
       +-------+
           |
           |
       +-------+
       |   2   | R(2)=R(1) + 512 = 1280
       |       | DAGRank(R(2)) = 5
       +-------+
           |
           |
       +-------+
       |   3   | R(3)=R(2) + 512 = 1792
       |       | DAGRank(R(3)) = 7
       +-------+
           |
           |
       +-------+
       |   4   | R(4)=R(3) + 512 = 2304
       |       | DAGRank(R(4)) = 9
       +-------+
           |
           |
       +-------+
       |   5   | R(5)=R(4) + 512 = 2816
       |       | DAGRank(R(5)) = 11
       +-------+
```

10.2. RPL Configuration

In addition to the Objective Function (OF), nodes in a multihop network MUST indicate the preferred mode of operation using the MOP field in DIO. Nodes not being able to operate in the specified mode of operation MUST only join as leaf nodes. RPL information and hop-by-hop extension headers MUST follow [RFC6553] and [RFC6554] specification. In the case that the packets formed at the LLN need to cross through intermediate routers, these MUST obey to the IP in IP encapsulation requirement specified by the [RFC6282] and [RFC2460]. RPI and RH3 extension headers and inner IP headers MUST be compressed according to [RFC6282].

10.2.1. Mode of Operation

Nodes implementing a minimal configuration and forming a multihop network, MUST support the non-storing ([RFC6550] Section 9.7) mode of operation. The storing ([RFC6550] Section 9.8) mode of operation SHOULD be supported by nodes with enough capabilities. Non-storing mode of operation is the default mode that a node selects when acting as a DAG root.

10.2.2. 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].

10.2.3. 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 (in the form 128*ETX), the recommendation for this document is to use PARENT_SWITCH_THRESHOLD equal to 640 if the metric being used is (3*ETX-2)*minHopRankIncrease, or a proportional value. This mechanism is suited to deal with parent hysteresis in both cases including routing parent and time source neighbor selection.

10.2.4. 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	640

11. Examples

Several examples are provided to illustrate the content of the packets used by the minimal configuration as proposed by this document. Each example follows the same structure presenting first a schematic header diagram, then the LSB stream of bytes that conform the header and finally a description of each of the IEs the form the packet. The packet formats are specific for the [IEEE802154-2012] revision and may vary in future releases of the IEEE standard. In case of differences between the packet content presented in this section and the [IEEE802154-2012], the later has presedence.

The MAC header fields are described in a specific order. All field formats in this examples are depicted in the order in which they are transmitted by the PHY, from left to right, where the leftmost bit is transmitted first in time. Bits within each field are numbered from 0 (leftmost and least significant) to k - 1 (rightmost and most significant), where the length of the field is k bits. Fields that are longer than a single octet are sent to the PHY in the order from the octet containing the lowest numbered bits to the octet containing the highest numbered bits, hence little endian ordering.

11.1. **Example 1.** Information Elements in EBs

Mandatory content for the EB as proposed by this draft. The example uses a slotframe of 101 slots. The following figure represents schematically the Header IE and Payload IE content of an EB.

Schematic representation of the IE header in an EB:

```
                    1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Len1 =   0  |Element ID=0x7e|0|    Len2 = 26        |GrpId=1|1|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Len3 =   6    |Sub ID = 0x1a|0|           ASN
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
              ASN                              | Join Priority |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Len4 = 0x01  |Sub ID = 0x1c|0| TT ID = 0x00  |   Len5 = 0x01
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |ID=0x9 |1| CH ID = 0x00  | Len6 = 0x0A   |Sub ID = 0x1b|0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   #SF = 0x01  | SF ID = 0x00  |   SF LEN = 0x65 (101 slots)   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| #Links = 0x01 |      SLOT OFFSET = 0x0000     |    CHANNEL
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  OFF  = 0x0000 |Link OPT = 0x0F|         NO MAC PAYLOAD
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Stream of bytes (in little-endian ordering) that derive
from the previous schematic header:

```
00 3F 1A 88 06 1A ASN#0 ASN#1 ASN#2 ASN#3 ASN#4 JP 01 1C 00
01 C8 00 0A 1B 01 00 65 00 01 00 00 00 00 0F
```

Description of the IE fields in the example:

#Header IE Header
Len1 = Header IE Length (0)
Element ID = 0x7e - termination IE indicating Payload IE coming next
Type 0

#Payload IE Header (MLME)
Len2 = Payload IE Len (26 Bytes)
GroupID = 1 MLME (Nested)
Type = 1

#MLME-SubIE TSCH Synchronization
Len3 = Length in bytes of the sub-IE payload (6 Bytes)
SubID = 0x1a (MLME-SubIE TSCH Synchronization)
Type = Short (0)
ASN = Absolute Sequence Number (5 Bytes)
Join Priority = 1 Byte

#MLME-SubIE TSCH TimeSlot

```
Len4 = Length in bytes of the sub-IE payload (1 Byte)
SubID = 0x1c (MLME-SubIE Timeslot)
Type = Short (0)
TimeSlot template ID = 0x00 (default)

#MLME-SubIE Ch. Hopping
Len5 = Length in bytes of the sub-IE payload (1 Byte)
SubID = 0x09 (MLME-SubIE Ch. Hopping)
Type = Long (1)
Channel Hopping Sequence ID = 0x00 (default)

#MLME-SubIE TSCH Slotframe and Link
Len6 = Length in bytes of the sub-IE payload (10 Bytes)
SubID = 0x1b (MLME-SubIE TSCH Slotframe and Link)
Type = Short (0)
Number of slotframes = 0x01
SlotFrame Handle = 0x00
SlotFrame Size = 101 slots (0x65)
Number of Links = 0x01
Timeslot = 0x0000 (2B)
Channel Offset = 0x0000 (2B)
Link Option = 0x0F (tx,rx,shared,timekeeping)
```

11.2. **Example 2.** Information Elements in EBs not using default timeslot template

Using a non-default timeslot template in EBs. Timeslot length set to 15ms instead of the 10ms default.

Schematic representation of the IE header in an EB:

```
                    1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Len1 =   0  |Element ID=0x7e|0|    Len2 = 53        |GrpId=1|1|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Len3 =   6    |Sub ID = 0x1a|0|           ASN
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
              ASN                              | Join Priority |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Len4 = 25    |Sub ID = 0x1c|0| TT ID = 0x01 | macTsCCAOffset
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   = 2700       |  macTsCCA = 128               | macTsTxOffset
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   = 3180       |  macTsRxOffset = 1680         | macTsRxAckDelay
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   = 1200       |  macTsTxAckDelay = 1500       | macTsRxWait
```

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   = 3300       |  macTsAckWait = 600           | macTsRxTx
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   = 192        |  macTsMaxAck  = 2400          | macTsMaxTx
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   = 4256       | macTsTimeslotLength = 15000   | Len5 = 0x01
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |ID=0x9 |1| CH ID = 0x00  | Len6 = 0x0A   | ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Stream of bytes (in little-endian ordering) that derive
from the previous schematic header:

```
00 3F 1A 88 06 1A ASN#0 ASN#1 ASN#2 ASN#3 ASN#4 JP 19 1C 01 8C 0A 80
00 6C 0C 90 06 B0 04 DC 05 E4 0C 58 02 C0 00 60 09 A0 10 98 3A 01 C8
00 0A ...
```

Description of the IE fields in the example:

#Header IE Header
Len1 = Header IE Length (none)
Element ID = 0x7e - termination IE indicating Payload IE coming next
Type 0

#Payload IE Header (MLME)
Len2 = Payload IE Len (53 Bytes)
GroupID = 1 MLME (Nested)
Type = 1

#MLME-SubIE TSCH Synchronization
Len3 = Length in bytes of the sub-IE payload (6 Bytes)
SubID = 0x1a (MLME-SubIE TSCH Synchronization)
Type = Short (0)
ASN = Absolute Sequence Number (5 Bytes)
Join Priority = 1 Byte

#MLME-SubIE TSCH TimeSlot
Len4 = Lenght in bytes of the sub-IE payload (25 Bytes)
SubID = 0x1c (MLME-SubIE Timeslot)
Type = Short (0)
TimeSlot template ID = 0x01 (non-default)

Example timeslot timming using 15ms timeslot.

IEEE802.15.4 TSCH parameter	Value (us)
tsCCAOffset	2700

tsCCA	128
tsTxOffset	3180
tsRxOffset	1680
tsRxAckDelay	1200
tsTxAckDelay	1500
tsRxWait	3300
tsAckWait	600
tsRxTx	192
tsMaxAck	2400
tsMaxTx	4256
Time Slot duration	15000

```
#MLME-SubIE Ch. Hopping
Len5 = Length in bytes of the sub-IE payload. (1 Byte)
SubID = 0x09 (MLME-SubIE Ch. Hopping)
Type = Long (1)
Channel Hopping Sequence ID = 0x00 (default)
```

11.3. **Example 3.** Information Elements in ACKs

Acknowledgement packets carry the ACK/NACK Time Correction IE (Header IE). The following example illustrates the IE format as specified in [IEEE802154-2012].

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Len1 =   2  |Element ID=0x1e|0|        Time Sync Info          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Stream of bytes (in little-endian ordering) that derive
from the previous schematic header:

02 0F TS#0 TS#1

Description of the IE fields in the example:

#Header IE Header
Len1 = Header IE Length (2 Bytes)
Element ID = 0x1e - ACK/NACK Time Correction IE
Type 0

11.4. **Example 4.** Auxiliary Security Header

The example illustrates content of the Auxiliary Security Header as mandated by this document, if security is enabled. Security Level in the example is set to ENC-MIC-32 (5).

```
                        1
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |L = 5|M=1|1|1|0|Key Index = IDX|
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Stream of bytes (in LSB format) that derive from the previous schematic header:

6D IDX#0

Description of the Security Auxiliary Header fields in the example:

#Security Control (1 byte)
L = Security Level ENC-MIC-32 (5)
M = Key Identifier Mode (0x01)
Frame Counter Suppression = 1 (omitting Frame Counter field)
Frame Counter Size = 1 (construct Nonce from 5 byte ASN)
Reserved = 0

#Key Identifier (1 byte)
Key Index = IDX (deployment-specific KeyIndex parameter that
            identifies the cryptographic key)

12. IANA Considerations

This document requests no immediate action by IANA.

13. Acknowledgments

The authors would like to acknowledge the guidance and input provided by Rene Struik, Pat Kinney, Michael Richardson, Tero Kivinen, Nicola Accettura, Malisa Vucinic and for the exhaustive and detailed review of the examples section to Simon Duquennoy, Guillaume Gaillard, Tengfei Chang and Jonathan Munoz. Also our acknowledge to the 6TiSCH Chairs Pascal Thubert and Thomas Watteyne for their guideance and advice.

14. References

14.1. Normative References

[RFC6719] Gnawali, O. and P. Levis, "The Minimum Rank with Hysteresis Objective Function", RFC 6719, DOI 10.17487/RFC6719, September 2012, <http://www.rfc-editor.org/info/rfc6719>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <http://www.rfc-editor.org/info/rfc6282>.

[RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, DOI 10.17487/RFC6554, March 2012, <http://www.rfc-editor.org/info/rfc6554>.

[RFC6553] Hui, J. and JP. Vasseur, "The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams", RFC 6553, DOI 10.17487/RFC6553, March 2012, <http://www.rfc-editor.org/info/rfc6553>.

[RFC6552] Thubert, P., Ed., "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6552, DOI 10.17487/RFC6552, March 2012, <http://www.rfc-editor.org/info/rfc6552>.

[RFC6551] Vasseur, JP., Ed., Kim, M., Ed., Pister, K., Dejean, N., and D. Barthel, "Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks", RFC 6551, DOI 10.17487/RFC6551, March 2012, <http://www.rfc-editor.org/info/rfc6551>.

[RFC6550] Winter, T., Ed., Thubert, P., Ed., 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, DOI 10.17487/RFC6550, March 2012, <http://www.rfc-editor.org/info/rfc6550>.

[RFC6206] Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko, "The Trickle Algorithm", RFC 6206, DOI 10.17487/RFC6206, March 2011, <http://www.rfc-editor.org/info/rfc6206>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <http://www.rfc-editor.org/info/rfc2460>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[IEEE802154-2012]
IEEE standard for Information Technology, "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 as amended by 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 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".

[decouto03high]
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.

14.2. Informative References

[RFC7554] Watteyne, T., Ed., Palattella, M., and L. Grieco, "Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement", RFC 7554, DOI 10.17487/RFC7554, May 2015, <http://www.rfc-editor.org/info/rfc7554>.

[RFC7102] Vasseur, JP., "Terms Used in Routing for Low-Power and Lossy Networks", RFC 7102, DOI 10.17487/RFC7102, January 2014, <http://www.rfc-editor.org/info/rfc7102>.

[RFC3610] Whiting, D., Housley, R., and N. Ferguson, "Counter with CBC-MAC (CCM)", RFC 3610, DOI 10.17487/RFC3610, September 2003, <http://www.rfc-editor.org/info/rfc3610>.

[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-05 (work in progress), July 2015.

[I-D.ietf-6tisch-6top-interface]
Wang, Q. and X. Vilajosana, "6TiSCH Operation Sublayer (6top) Interface", draft-ietf-6tisch-6top-interface-04 (work in progress), July 2015.

14.3. External Informative References

[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.

[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