Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e (6TiSCH) Configuration

Abstract

   This document describes a minimal mode of operation for an IPv6 over
   the TSCH mode of IEEE 802.15.4e (6TiSCH) network. This minimal mode
   of operation specifies the baseline set of protocols that need to be
   supported and the recommended configurations and modes of operation
   sufficient to enable a 6TiSCH functional network. 6TiSCH provides
   IPv6 connectivity over a Time-Slotted Channel Hopping (TSCH) mesh
   composed of IEEE Std 802.15.4 TSCH links. This minimal mode uses a
   collection of protocols with the respective configurations, including
   the IPv6 Low-Power Wireless Personal Area Network (6LoWPAN)
   framework, enabling interoperable IPv6 connectivity over IEEE Std
   802.15.4 TSCH. This minimal configuration provides the necessary
   bandwidth for network and security bootstrapping and defines the
   proper link between the IETF protocols that interface to IEEE Std
   802.15.4 TSCH. This minimal mode of operation should be implemented
   by all 6TiSCH-compliant devices.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

   A 6TiSCH network provides IPv6 connectivity [RFC2460] over a Time-
   Slotted Channel Hopping (TSCH) mesh [RFC7554] composed of IEEE Std
   802.15.4 TSCH links [IEEE.802.15.4]. IPv6 connectivity is obtained
   by the use of the 6LoWPAN framework ([RFC4944], [RFC6282],
   [RFC8025],[RFC8138], and [RFC6775]), RPL [RFC6550], and the RPL
   Objective Function 0 (OF0) [RFC6552].

   This specification defines operational parameters and procedures for
   a minimal mode of operation to build a 6TiSCH network. Any 6TiSCH-
   compliant device should implement this mode of operation. This
   operational parameter configuration provides the necessary bandwidth
   for nodes to bootstrap the network. The bootstrap process includes
   initial network configuration and security bootstrapping. In this
   specification, the 802.15.4 TSCH mode, the 6LoWPAN framework, RPL
   [RFC6550], and the RPL Objective Function 0 (OF0) [RFC6552] are used
   unmodified. Parameters and particular operations of TSCH are
   specified to guarantee interoperability between nodes in a 6TiSCH
   network.

   In a 6TiSCH network, nodes follow a communication schedule as per
   802.15.4 TSCH. Nodes learn the communication schedule upon joining
   the network. When following this specification, the learned schedule
   is the same for all nodes and does not change over time. Future
   specifications may define mechanisms for dynamically managing the
   communication schedule. Dynamic scheduling solutions are out of
   scope of this document.

   IPv6 addressing and compression are achieved by the 6LoWPAN
   framework. The framework includes [RFC4944], [RFC6282], [RFC8025],
   the 6LoWPAN Routing Header dispatch [RFC8138] for addressing and
   header compression, and [RFC6775] for Duplicate Address Detection
   (DAD) and address resolution.

   More advanced work is expected in the future to complement the
   minimal configuration with dynamic operations that can adapt the
   schedule to the needs of the traffic at run time.

2. Requirements Language

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and
   "OPTIONAL" in this document are to be interpreted as described in BCP
   14 [RFC2119] [RFC8174] when, and only when, they appear in all
   capitals, as shown here.

3. Terminology

   This document uses terminology from [TERMS-6TiSCH]. The following
   concepts are used in this document:

   802.15.4: We use "802.15.4" as a short version of "IEEE Std
      802.15.4" in this document.

   SFD: Start of Frame Delimiter

   RX: Reception

   TX: Transmission

   IE: Information Element

   EB: Enhanced Beacon

   ASN: Absolute Slot Number

   Join Metric: Field in the TSCH Synchronization IE representing the
      topological distance between the node sending the EB and the PAN
      coordinator.

   PAN: Personal Area Network

   MLME: MAC Layer Management Entity

4. IEEE Std 802.15.4 Settings

   An implementation compliant with this specification MUST implement
   IEEE Std 802.15.4 [IEEE.802.15.4] in Time-Slotted Channel Hopping
   (TSCH) mode.

   The remainder of this section details the RECOMMENDED TSCH settings,
   which are summarized in Figure 1. Any of the properties marked in
   the EB column are announced in the EBs the nodes send [IEEE.802.15.4]
   and learned by those joining the network. Changing their value means
   changing the contents of the EB.

   In case of discrepancy between the values in this specification and
   IEEE Std 802.15.4 [IEEE.802.15.4], the IEEE standard has precedence.

```
+-----------------------------+-----------------------------+---+
|          Property           |     Recommended Setting     |EB*|
+-----------------------------+-----------------------------+---+
| Slotframe Size              | Tunable. Trades off         | X |
|                             | bandwidth against energy.   |   |
+-----------------------------+-----------------------------+---+
| Number of scheduled cells** | 1                           | X |
| (active)                    | Timeslot       0x0000       |   |
|                             | Channel Offset  0x0000      |   |
|                             | Link Options = (TX Link = 1,|   |
|                             | RX Link = 1, Shared Link = 1,|  |
|                             | Timekeeping = 1)            |   |
+-----------------------------+-----------------------------+---+
| Number of unscheduled cells | All remaining cells in the  | X |
| (off)                       | slotframe.                  |   |
+-----------------------------+-----------------------------+---+
| Max Number MAC retransmissions | 3 (4 transmission attempts) |  |
+-----------------------------+-----------------------------+---+
| Timeslot template           | IEEE Std 802.15.4 default    | X |
|                             | (macTimeslotTemplateId=0)    |   |
+-----------------------------+-----------------------------+---+
| Enhanced Beacon Period      | Tunable. Trades off join    |   |
| (EB_PERIOD)                 | time against energy.        |   |
+-----------------------------+-----------------------------+---+
| Number used frequencies     | IEEE Std 802.15.4 default    | X |
| (2.4 GHz O-QPSK PHY)        | (16)                         |   |
+-----------------------------+-----------------------------+---+
| Channel Hopping sequence    | IEEE Std 802.15.4 default    | X |
| (2.4 GHz O-QPSK PHY)        | (macHoppingSequenceID = 0)   |   |
+-----------------------------+-----------------------------+---+
```

   * An "X" in this column means this property's value is announced
     in the EB; hence, a new node learns it when joining.
  ** This cell LinkType is set to ADVERTISING.

          Figure 1: Recommended IEEE Std 802.15.4 TSCH Settings

4.1. TSCH Schedule

   This minimal mode of operation uses a single slotframe. The TSCH
   slotframe is composed of a tunable number of timeslots. The
   slotframe size (i.e., the number of timeslots it contains) trades off
   bandwidth for energy consumption. The slotframe size needs to be
   tuned; the way of tuning it is out of scope of this specification.
   The slotframe size is announced in the EB. The RECOMMENDED value for
   the slotframe handle (macSlotframeHandle) is 0x00. An implementation
   MAY choose to use a different slotframe handle, for example, to add
   other slotframes with higher priority. The use of other slotframes
   is out of the scope of this document.

There is only a single scheduled cell in the slotframe. This cell
MAY be scheduled at any slotOffset/channelOffset within the
slotframe. The location of that cell in the schedule is announced in
the EB. The LinkType of the scheduled cell is ADVERTISING to allow
EBs to be sent on it.

Figure 2 shows an example of a slotframe of length 101 timeslots,
resulting in a radio duty cycle below 0.99%.
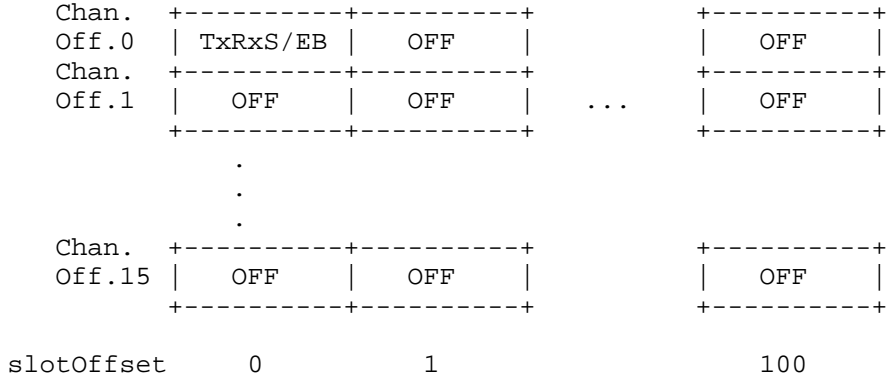
```
    Chan.  +----------+----------+          +----------+
    Off.0  | TxRxS/EB |   OFF    |          |   OFF    |
    Chan.  +----------+----------+          +----------+
    Off.1  |   OFF    |   OFF    |  ...     |   OFF    |
           +----------+----------+          +----------+
                 .
                 .
                 .
    Chan.  +----------+----------+          +----------+
    Off.15 |   OFF    |   OFF    |          |   OFF    |
           +----------+----------+          +----------+

slotOffset     0          1                    100

EB:  Enhanced Beacon
Tx:  Transmit
Rx:  Receive
S:   Shared
OFF: Unscheduled by this specification
```

           Figure 2: Example Slotframe of Length 101 Timeslots

A node MAY use the scheduled cell to transmit/receive all types of
link-layer frames. EBs are sent to the link-layer broadcast address
and are not acknowledged. Data frames are sent unicast and are
acknowledged by the receiving neighbor.

All remaining cells in the slotframe are unscheduled. Dynamic
scheduling solutions may be defined in the future that schedule those
cells. One example is the 6top Protocol (6P) [PROTO-6P]. Dynamic
scheduling solutions are out of scope of this document.

The default values of the TSCH timeslot template (defined in
Section 8.4.2.2.3 of [IEEE.802.15.4]) and channel hopping sequence
(defined in Section 6.2.10 of [IEEE.802.15.4]) SHOULD be used. A
node MAY use different values by properly announcing them in its EB.

4.2. Cell Options

   In the scheduled cell, a node transmits if there is a packet to
   transmit and listens otherwise (both "TX" and "RX" bits are set).
   When a node transmits, requesting a link-layer acknowledgment per
   [IEEE.802.15.4], and does not receive the requested acknowledgement,
   it uses a back-off mechanism to resolve possible collisions ("Shared"
   bit is set). A node joining the network maintains time
   synchronization to its initial time-source neighbor using that cell
   ("Timekeeping" bit is set).

   This translates into a Link Option for this cell:

      b0 = TX Link = 1 (set)
      b1 = RX Link = 1 (set)
      b2 = Shared Link = 1 (set)
      b3 = Timekeeping = 1 (set)
      b4 = Priority = 0 (clear)
      b5-b7 = Reserved = 0 (clear)

4.3. Retransmissions

   Per Figure 1, the RECOMMENDED maximum number of link-layer
   retransmissions is 3. This means that, for packets requiring an
   acknowledgment, if none are received after a total of 4 attempts, the
   transmission is considered failed and the link layer MUST notify the
   upper layer. Packets not requiring an acknowledgment (including EBs)
   are not retransmitted.

4.4. Timeslot Timing

   Per Figure 1, the RECOMMENDED timeslot template is the default one
   (macTimeslotTemplateId=0) defined in [IEEE.802.15.4].

4.5. Frame Contents

   [IEEE.802.15.4] defines the format of frames. Through a set of
   flags, [IEEE.802.15.4] allows for several fields to be present (or
   not), to have different lengths, and to have different values. This
   specification details the RECOMMENDED contents of 802.15.4 frames,
   while strictly complying with [IEEE.802.15.4].

4.5.1. IEEE Std 802.15.4 Header

   The Frame Version field MUST be set to 0b10 (Frame Version 2). The
   Sequence Number field MAY be elided.

The EB Destination Address field MUST be set to 0xFFFF (short
broadcast address). The EB Source Address field SHOULD be set as the
node's short address if this is supported. Otherwise, the long
address MUST be used.

The PAN ID Compression bit SHOULD indicate that the Source PAN ID is
"Not Present" and the Destination PAN ID is "Present". The value of
the PAN ID Compression bit is specified in Table 7-2 of the IEEE Std
802.15.4-2015 specification and depends on the type of the
destination and source link-layer addresses (e.g., short, extended,
not present).

Nodes follow the reception and rejection rules as per Section 6.7.2
of [IEEE.802.15.4].

The nonce is formatted according to [IEEE.802.15.4]. In the IEEE Std
802.15.4 specification [IEEE.802.15.4], nonce generation is described
in Section 9.3.2.2, and byte ordering is described in Section 9.3.1,
Annex B.2, and Annex B.2.2.

4.5.2. Enhanced Beacon Frame

After booting, a TSCH node starts in an unsynchronized, unjoined
state. Initial synchronization is achieved by listening for EBs.
EBs from multiple networks may be heard. Many mechanisms exist for
discrimination between networks, the details of which are out of
scope.

The IEEE Std 802.15.4 specification does not define how often EBs are
sent, nor their contents [IEEE.802.15.4]. In a minimal TSCH
configuration, a node SHOULD send an EB every EB_PERIOD. Tuning
EB_PERIOD allows a trade-off between joining time and energy
consumption.

EBs should be used to obtain information about local networks and to
synchronize ASN and time offset of the specific network that the node
decides to join. Once joined to a particular network, a node MAY
choose to continue to listen for EBs, to gather more information
about other networks, for example. During the joining process,
before secure connections to time parents have been created, a node
MAY maintain synchronization using EBs. [RFC7554] discusses
different time synchronization approaches.

The IEEE Std 802.15.4 specification requires EBs to be sent in order
to enable nodes to join the network. The EBs SHOULD carry the
Information Elements (IEs) listed below [IEEE.802.15.4].

TSCH Synchronization IE: Contains synchronization information such
   as ASN and Join Metric. The value of the Join Metric field is
   discussed in Section 6.1.

TSCH Timeslot IE: Contains the timeslot template identifier. This
   template is used to specify the internal timing of the timeslot.
   This specification RECOMMENDS the default timeslot template.

Channel Hopping IE: Contains the channel hopping sequence
   identifier. This specification RECOMMENDS the default channel
   hopping sequence.

TSCH Slotframe and Link IE: Enables joining nodes to learn the
   initial schedule to be used as they join the network. This
   document RECOMMENDS the use of a single cell.

If a node strictly follows the recommended setting from Figure 1, the
EB it sends has the exact same contents as an EB it received when
joining, except for the Join Metric field in the TSCH Synchronization
IE.

When a node has already joined a network (i.e., it has received an
EB) synchronized to the EB sender and configured its schedule
following this specification, the node SHOULD ignore subsequent EBs
that try to change the configured parameters. This does not preclude
listening to EBs from other networks.

4.5.3. Acknowledgment Frame

   Per [IEEE.802.15.4], each acknowledgment contains an ACK/NACK Time
   Correction IE.

4.6. Link-Layer Security

   When securing link-layer frames, link-layer frames MUST be secured by
   the link-layer security mechanisms defined in IEEE Std 802.15.4
   [IEEE.802.15.4]. Link-layer authentication MUST be applied to the
   entire frame, including the 802.15.4 header. Link-layer encryption
   MAY be applied to 802.15.4 Payload IEs and the 802.15.4 payload.

   This specification assumes the existence of two cryptographic keys:

      Key K1 is used to authenticate EBs. EBs MUST be authenticated
      only (no encryption); their contents are defined in Section 4.5.2.

      Key K2 is used to authenticate and encrypt DATA and ACKNOWLEDGMENT
      frames.

   These keys can be pre-configured or learned during a key distribution
   phase. Key distribution mechanisms are defined, for example, in
   [SEC-6TISCH] and [SEC-JOIN-6TISCH]. Key distribution is out of scope
   of this document.

   The behavior of a Joining Node (JN) is different depending on which
   key(s) are pre-configured:

      If both keys K1 and K2 are pre-configured, the JN does not rely on
      a key distribution phase to learn K1 or K2.

      If key K1 is pre-configured but not key K2, the JN authenticates
      EBs using K1 and relies on the key distribution phase to learn K2.

      If neither key K1 nor key K2 is pre-configured, the JN accepts EBs
      as defined in Section 6.3.1.2 of IEEE Std 802.15.4
      [IEEE.802.15.4], i.e., they are passed forward even "if the status
      of the unsecuring process indicated an error". The JN then runs
      the key distribution phase to learn K1 and K2. During that
      process, the node that JN is talking to uses the secExempt
      mechanism (see Section 9.2.4 of [IEEE.802.15.4]) to process frames
      from JN. Once the key distribution phase is done, the node that
      has installed secExempts for the JN MUST clear the installed
      exception rules.

   In the event of a network reset, the new network MUST either use new
   cryptographic keys or ensure that the ASN remains monotonically
   increasing.

5. RPL Settings

   In a multi-hop topology, the RPL routing protocol [RFC6550] MAY be
   used.

5.1. Objective Function

   If RPL is used, nodes MUST implement the RPL Objective Function Zero
   (OF0) [RFC6552].

5.1.1. Rank Computation

   The Rank computation is described in Section 4.1 of [RFC6552]. A
   node's Rank (see Figure 4 for an example) is computed by the
   following equations:

      R(N) = R(P) + rank_increment

      rank_increment = (Rf*Sp + Sr) * MinHopRankIncrease

Figure 3 lists the OF0 parameter values that MUST be used if RPL is
used.

```
+----------------------+------------------------------------+
|    OF0 Parameters    |               Value                |
+----------------------+------------------------------------+
| Rf                   |                                  1 |
+----------------------+------------------------------------+
| Sp                   |                          (3*ETX)-2 |
+----------------------+------------------------------------+
| Sr                   |                                  0 |
+----------------------+------------------------------------+
| MinHopRankIncrease   | DEFAULT_MIN_HOP_RANK_INCREASE (256) |
+----------------------+------------------------------------+
| MINIMUM_STEP_OF_RANK |                                  1 |
+----------------------+------------------------------------+
| MAXIMUM_STEP_OF_RANK |                                  9 |
+----------------------+------------------------------------+
| ETX limit to select  |                                  3 |
| a parent             |                                    |
+----------------------+------------------------------------+
```

Figure 3: OF0 Parameters

The step_of_rank (Sp) uses the Expected Transmission Count (ETX)
[RFC6551].

An implementation MUST follow OF0's normalization guidance as
discussed in Sections 1 and 4.1 of [RFC6552]. Sp SHOULD be
calculated as (3*ETX)-2. The minimum value of Sp
(MINIMUM_STEP_OF_RANK) indicates a good quality link. The maximum
value of Sp (MAXIMUM_STEP_OF_RANK) indicates a poor quality link.
The default value of Sp (DEFAULT_STEP_OF_RANK) indicates an average
quality link. Candidate parents with ETX greater than 3 SHOULD NOT
be selected. This avoids having ETX values on used links that are
larger that the maximum allowed transmission attempts.

5.1.2. Rank Computation Example

   This section illustrates the use of OF0 (see Figure 4). We have:

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

```
     +-------+
     |   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
     +-------+
```
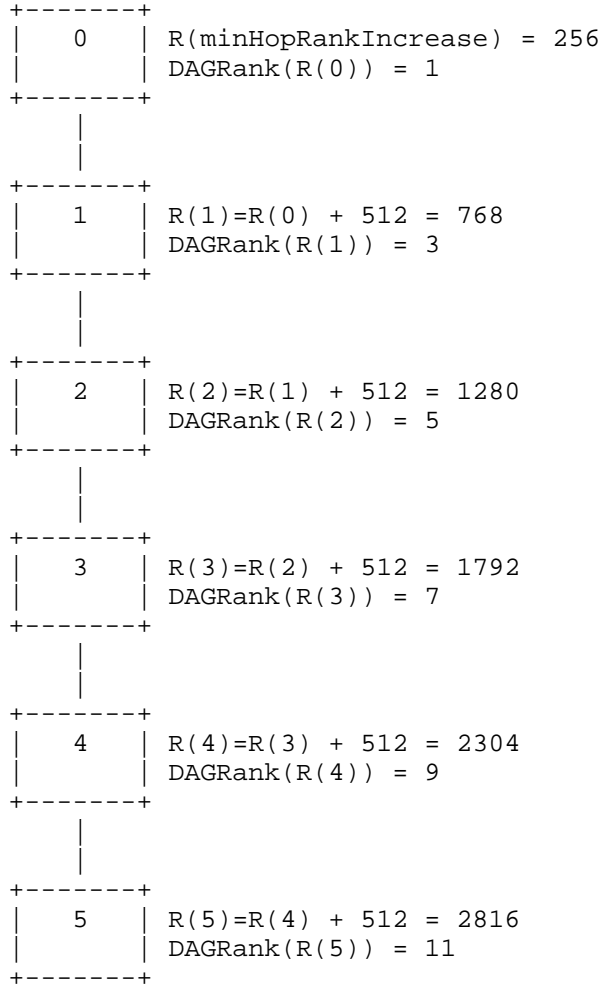
      Figure 4: Rank computation example for a 5-hop network where
                numTx=100 and numTxAck=75 for all links.

5.2. Mode of Operation

   When RPL is used, nodes MUST implement the non-storing mode of
   operation (see Section 9.7 of [RFC6550]). The storing mode of
   operation (see Section 9.8 of [RFC6550]) SHOULD be implemented by
   nodes with enough capabilities. Nodes not implementing RPL MUST join
   as leaf nodes.

5.3. Trickle Timer

   RPL signaling messages such as DODAG Information Objects (DIOs) are
   sent using the Trickle algorithm (see Section 8.3.1 of [RFC6550] and
   Section 4.2 of [RFC6206]). For this specification, the Trickle timer
   MUST be used with the RPL-defined default values (see Section 8.3.1
   of [RFC6550]).

5.4. Packet Contents

   RPL information and hop-by-hop extension headers MUST follow
   [RFC6553] and [RFC6554]. For cases in which the packets formed at
   the Low-Power and Lossy Network (LLN) need to cross through
   intermediate routers, these MUST follow the IP-in-IP encapsulation
   requirement specified by [RFC6282] and [RFC2460]. Routing extension
   headers such as RPL Packet Information (RPI) [RFC6550] and Source
   Routing Header (SRH) [RFC6554], and outer IP headers in case of
   encapsulation, MUST be compressed according to [RFC8138] and
   [RFC8025].

6. Network Formation and Lifetime

6.1. Value of the Join Metric Field

   The Join Metric of the TSCH Synchronization IE in the EB MUST be
   calculated based on the routing metric of the node, normalized to a
   value between 0 and 255. A lower value of the Join Metric indicates
   the node sending the EB is topologically "closer" to the root of the
   network. A lower value of the Join Metric hence indicates higher
   preference for a joining node to synchronize to that neighbor.

   In case the network uses RPL, the Join Metric of any node (including
   the Directed Acyclic Graph (DAG) root) MUST be set to
   DAGRank(rank)-1. According to Section 5.1.1, DAGRank(rank(0)) = 1.
   DAGRank(rank(0))-1 = 0 is compliant with 802.15.4's requirement of
   having the root use Join Metric = 0.

   In case the network does not use RPL, the Join Metric value MUST
   follow the rules specified by [IEEE.802.15.4].

6.2. Time-Source Neighbor Selection

   When a node joins a network, it may hear EBs sent by different nodes
   already in the network. The decision of which neighbor to
   synchronize to (e.g., which neighbor becomes the node's initial time-
   source neighbor) is implementation specific. For example, after
   having received the first EB, a node MAY listen for at most
   MAX_EB_DELAY seconds until it has received EBs from
   NUM_NEIGHBOURS_TO_WAIT distinct neighbors. Recommended values for
   MAX_EB_DELAY and NUM_NEIGHBOURS_TO_WAIT are defined in Figure 5.
   When receiving EBs from distinct neighbors, the node MAY use the Join
   Metric field in each EB to select the initial time-source neighbor,
   as described in Section 6.3.6 of IEEE Std 802.15.4 [IEEE.802.15.4].

   At any time, a node MUST maintain synchronization to at least one
   time-source neighbor. A node's time-source neighbor MUST be chosen
   among the neighbors in its RPL routing parent set when RPL is used.
   In the case a node cannot maintain connectivity to at least one time-
   source neighbor, the node looses synchronization and needs to join
   the network again.

6.3. When to Start Sending EBs

   When a RPL node joins the network, it MUST NOT send EBs before having
   acquired a RPL Rank to avoid inconsistencies in the time
   synchronization structure. This applies to other routing protocols
   with their corresponding routing metrics. As soon as a node acquires
   routing information (e.g., a RPL Rank, see Section 5.1.1), it SHOULD
   start sending EBs.

6.4. Hysteresis

   Per [RFC6552] and [RFC6719], the specification RECOMMENDS the use of
   a boundary value (PARENT_SWITCH_THRESHOLD) to avoid constant changes
   of the parent when ranks are compared. When evaluating a parent that
   belongs to a smaller path cost than the current minimum path, the
   candidate node is selected as the 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. Per [RFC6719], the PARENT_SWITCH_THRESHOLD
   SHOULD be set to 192 when the 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 deals
   with hysteresis both for routing parent and time-source neighbor
   selection.

7. Implementation Recommendations

7.1. Neighbor Table

   The exact format of the neighbor table is implementation specific.
   The RECOMMENDED per-neighbor information is (taken from the [openwsn]
   implementation):

   identifier: Identifier(s) of the neighbor (e.g., EUI-64).

   numTx: Number of link-layer transmission attempts to that
               neighbor.

   numTxAck: Number of transmitted link-layer frames that have been
               link-layer acknowledged by that neighbor.

   numRx: Number of link-layer frames received from that neighbor.

   timestamp: When the last frame was 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.

   routing metric: The RPL Rank of that neighbor, for example.

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

7.2. Queues and Priorities

   The IEEE Std 802.15.4 specification [IEEE.802.15.4] does not define
   the use of queues to handle upper-layer data (either application or
   control data from upper layers). The following rules are
   RECOMMENDED:

      A node is configured to keep in the queues a configurable number
      of upper-layer packets per link (default NUM_UPPERLAYER_PACKETS)
      for a configurable time that should cover the join process
      (default MAX_JOIN_TIME).

      Frames generated by the 802.15.4 layer (including EBs) are queued
      with a priority higher than frames coming from higher layers.

      A frame type BEACON is queued with higher priority than frame
      types DATA.

7.3. Recommended Settings

   Figure 5 lists RECOMMENDED values for the settings discussed in this
   specification.

```
       +-------------------------+------------------+
       | Parameter               | RECOMMENDED Value |
       +-------------------------+------------------+
       | MAX_EB_DELAY            |              180 |
       +-------------------------+------------------+
       | NUM_NEIGHBOURS_TO_WAIT  |                2 |
       +-------------------------+------------------+
       | PARENT_SWITCH_THRESHOLD |              640 |
       +-------------------------+------------------+
       | NUM_UPPERLAYER_PACKETS  |                1 |
       +-------------------------+------------------+
       | MAX_JOIN_TIME           |              300 |
       +-------------------------+------------------+
```

                   Figure 5: Recommended Settings

8. Security Considerations

   This document is concerned only with link-layer security.

   By their nature, many Internet of Things (IoT) networks have nodes in
   physically vulnerable locations. We should assume that nodes will be
   physically compromised, their memories examined, and their keys
   extracted. Fixed secrets will not remain secret. This impacts the
   node-joining process. Provisioning a network with a fixed link key
   K2 is not secure. For most applications, this implies that there
   will be a joining phase during which some level of authorization will
   be allowed for nodes that have not been authenticated. Details are
   out of scope, but the link layer must provide some flexibility here.

   If an attacker has obtained K1, it can generate fake EBs to attack a
   whole network by sending authenticated EBs. The attacker can cause
   the joining node to initiate the joining process to the attacker. In
   the case that the joining process includes authentication and
   distribution of a K2, then the joining process will fail and the JN
   will notice the attack. If K2 is also compromised, the JN will not
   notice the attack and the network will be compromised.

   Even if an attacker does not know the value of K1 and K2
   (Section 4.6), it can still generate fake EB frames authenticated
   with an arbitrary key. Here we discuss the impact these fake EBs can
   have, depending on what key(s) are pre-provisioned.

If both K1 and K2 are pre-provisioned; a joining node can
distinguish legitimate from fake EBs and join the legitimate
network. The fake EBs have no impact.

The same holds if K1 is pre-provisioned but not K2.

If neither K1 nor K2 is pre-provisioned, a joining node may
mistake a fake EB for a legitimate one and initiate a joining
process to the attacker. That joining process will fail, as the
joining node will not be able to authenticate the attacker during
the security handshake. This will force the joining node to start
over listening for an EB. So while the joining node never joins
the attacker, this costs the joining node time and energy and is a
vector of attack.

Choosing what key(s) to pre-provision needs to balance the different
discussions above.

Once the joining process is over, the node that has joined can
authenticate EBs (it knows K1). This means it can process their
contents and use EBs for synchronization.

ASN provides a nonce for security operations in a slot. Any re-use
of ASN with a given key exposes information about encrypted packet
contents and risks replay attacks. Replay attacks are prevented
because, when the network resets, either the new network uses new
cryptographic key(s) or ensures that the ASN increases monotonically
(Section 4.6).

Maintaining accurate time synchronization is critical for network
operation. Accepting timing information from unsecured sources MUST
be avoided during normal network operation, as described in
Section 4.5.2. During joining, a node may be susceptible to timing
attacks before key K1 and K2 are learned. During network operation,
a node MAY maintain statistics on time updates from neighbors and
monitor for anomalies.

Denial-of-Service (DoS) attacks at the Media Access Control (MAC)
layer in an LLN are easy to achieve simply by Radio Frequency (RF)
jamming. This is the base case against which more sophisticated DoS
attacks should be judged. For example, sending fake EBs announcing a
very low Join Metric may cause a node to waste time and energy trying
to join a fake network even when legitimate EBs are being heard.
Proper join security will prevent the node from joining the false
flag, but by then the time and energy will have been wasted.
However, the energy cost to the attacker would be lower and the

energy cost to the joining node would be higher if the attacker simply sent loud short packets in the middle of any valid EB that it hears.

ACK reception probability is less than 100% due to changing channel conditions and unintentional or intentional jamming. This will cause the sending node to retransmit the same packet until it is acknowledged or a retransmission limit is reached. Upper-layer protocols should take this into account, possibly using a sequence number to match retransmissions.

The 6TiSCH layer SHOULD keep track of anomalous events and report them to a higher authority. For example, EBs reporting low Join Metrics for networks that cannot be joined, as described above, may be a sign of attack. Additionally, in normal network operation, message integrity check failures on packets with a valid Cyclic Redundancy Check (CRC) will occur at a rate on the order of once per million packets. Any significant deviation from this rate may be a sign of a network attack. Along the same lines, time updates in ACKs or EBs that are inconsistent with the MAC-layer's sense of time and its own plausible time-error drift rate may also be a result of network attack.

9. IANA Considerations

   This document does not require any IANA actions.

10. References

10.1. Normative References

   [IEEE.802.15.4]
             IEEE, "IEEE Standard for Low-Rate Wireless Networks",
             IEEE 802.15.4,
             <http://ieeexplore.ieee.org/document/7460875/>.

   [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>.

   [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>.

   [RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler,
              "Transmission of IPv6 Packets over IEEE 802.15.4
              Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007,
              <http://www.rfc-editor.org/info/rfc4944>.

   [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>.

   [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>.

   [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>.

   [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>.

   [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>.

   [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>.

   [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>.

   [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>.

   [RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C.
              Bormann, "Neighbor Discovery Optimization for IPv6 over
              Low-Power Wireless Personal Area Networks (6LoWPANs)",
              RFC 6775, DOI 10.17487/RFC6775, November 2012,
              <http://www.rfc-editor.org/info/rfc6775>.

   [RFC8025] Thubert, P., Ed. and R. Cragie, "IPv6 over Low-Power
              Wireless Personal Area Network (6LoWPAN) Paging Dispatch",
              RFC 8025, DOI 10.17487/RFC8025, November 2016,
              <http://www.rfc-editor.org/info/rfc8025>.

   [RFC8138] Thubert, P., Ed., Bormann, C., Toutain, L., and R. Cragie,
              "IPv6 over Low-Power Wireless Personal Area Network
              (6LoWPAN) Routing Header", RFC 8138, DOI 10.17487/RFC8138,
              April 2017, <http://www.rfc-editor.org/info/rfc8138>.

   [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
              2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
              May 2017, <http://www.rfc-editor.org/info/rfc8174>.

10.2. Informative References

   [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, Volume 23 Issue 5, pages 480-493, DOI
              10.1002/ett.2558, August 2012.

   [PROTO-6P]
              Wang, Q., Vilajosana, X., and T. Watteyne, "6top Protocol
              (6P)", Work in Progress, draft-ietf-6tisch-6top-protocol-
              05, May 2017.

   [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>.

   [SEC-6TISCH]
              Vucinic, M., Simon, J., Pister, K., and M. Richardson,
              "Minimal Security Framework for 6TiSCH", Work in
              Progress, draft-ietf-6tisch-minimal-security-02, March
              2017.

   [SEC-JOIN-6TISCH]
              Richardson, M., "6tisch Secure Join protocol", Work in
              Progress, draft-ietf-6tisch-dtsecurity-secure-join-01,
              February 2017.

   [TERMS-6TiSCH]
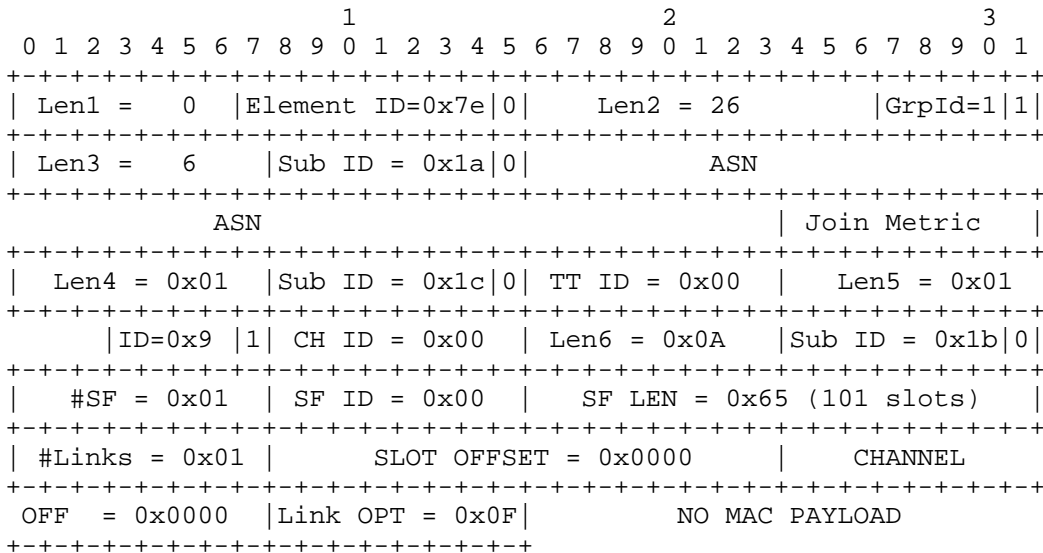   Palattella, M., Thubert, P., Watteyne, T., and Q. Wang,
              "Terminology in IPv6 over the TSCH mode of IEEE
              802.15.4e", Work in Progress, draft-ietf-6tisch-
              terminology-08, December 2016.

Appendix A. Examples

   This section contains several example packets. Each example contains
   (1) a schematic header diagram, (2) the corresponding bytestream, and
   (3) a description of each of the IEs that form the packet. Packet
   formats are specific for the [IEEE.802.15.4] revision and may vary in
   future releases of the IEEE standard. In case of differences between
   the packet content presented in this section and [IEEE.802.15.4], the
   latter has precedence.

   The MAC header fields are described in a specific order. All field
   formats in this example are depicted in the order in which they are
   transmitted, from left to right, where the leftmost bit is
   transmitted first. 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 (little endian).

A.1. Example: EB with Default Timeslot Template

```
                     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 Metric   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   Bytestream:

       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 IEs:

        #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)
            Group ID = 1 MLME (Nested)
            Type = 1

        #MLME-SubIE TSCH Synchronization
            Len3 = Length in bytes of the sub-IE payload (6 bytes)
            Sub-ID = 0x1a (MLME-SubIE TSCH Synchronization)
            Type = Short (0)
            ASN = Absolute Sequence Number (5 bytes)
            Join Metric = 1 byte

        #MLME-SubIE TSCH Timeslot
            Len4 = Length in bytes of the sub-IE payload (1 byte)
            Sub-ID = 0x1c (MLME-SubIE Timeslot)
            Type = Short (0)
            Timeslot template ID = 0x00 (default)

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

        #MLME-SubIE TSCH Slotframe and Link
            Len6 = Length in bytes of the sub-IE payload (10 bytes)
            Sub-ID = 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 (Cells) = 0x01
            Timeslot = 0x0000 (2B)
            Channel Offset = 0x0000 (2B)
            Link Options = 0x0F
            (TX Link = 1, RX Link = 1, Shared Link = 1,
             Timekeeping = 1)

A.2. Example: EB with Custom Timeslot Template

   Using a custom timeslot template in EBs: setting timeslot length to
   15 ms.

```
                         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 Metric   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  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   | ...
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   Bytestream:

   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 IEs:

       #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)
    Group ID = 1 MLME (Nested)
    Type = 1

#MLME-SubIE TSCH Synchronization
    Len3 = Length in bytes of the sub-IE payload (6 bytes)
    Sub-ID = 0x1a (MLME-SubIE TSCH Synchronization)
    Type = Short (0)
    ASN  = Absolute Sequence Number (5 bytes)
    Join Metric = 1 byte

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

    The 15 ms timeslot announced:
```
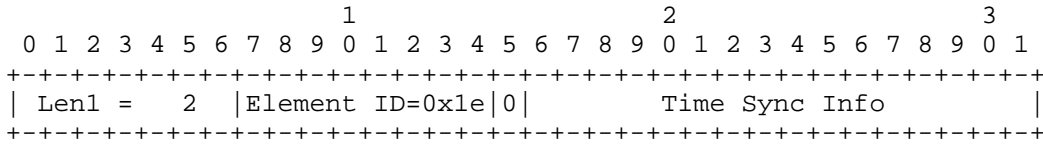
IEEE 802.15.4 TSCH parameter	Value (us)
macTsCCAOffset	2700
macTsCCA	128
macTsTxOffset	3180
macTsRxOffset	1680
macTsRxAckDelay	1200
macTsTxAckDelay	1500
macTsRxWait	3300
macTsAckWait	600
macTsRxTx	192
macTsMaxAck	2400
macTsMaxTx	4256
macTsTimeslotLength	15000

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

A.3. Example: Link-layer Acknowledgment

   Enhanced Acknowledgment packets carry the Time Correction IE (Header
   IE).

```
                          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         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
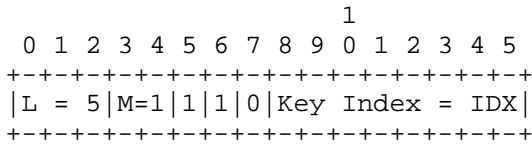
   Bytestream:

```
        02 0F TS#0 TS#1
```

   Description of the IEs:

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

A.4. Example: Auxiliary Security Header

   802.15.4 Auxiliary Security Header with the Security Level set to
   ENC-MIC-32.

```
                      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|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   Bytestream:

```
        6D IDX#0
```

   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)
          ASN in Nonce = 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)

Acknowledgments

   The authors acknowledge the guidance and input from Rene Struik, Pat
   Kinney, Michael Richardson, Tero Kivinen, Nicola Accettura, Malisa
   Vucinic, and Jonathan Simon. Thanks to Charles Perkins, Brian E.
   Carpenter, Ralph Droms, Warren Kumari, Mirja Kuehlewind, Ben
   Campbell, Benoit Claise, and Suresh Krishnan for the exhaustive and
   detailed reviews. Thanks to Simon Duquennoy, Guillaume Gaillard,
   Tengfei Chang, and Jonathan Munoz for the detailed review of the
   examples section. Thanks to 6TiSCH co-chair Pascal Thubert for his
   guidance and advice.

Authors' Addresses

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

   Email: xvilajosana@uoc.edu

   Kris Pister
   University of California Berkeley
   512 Cory Hall
   Berkeley, California 94720
   United States of America

   Email: pister@eecs.berkeley.edu

   Thomas Watteyne
   Analog Devices
   32990 Alvarado-Niles Road, Suite 910
   Union City, CA 94587
   United States of America

   Email: twatteyne@linear.com