Exegin Technologies Limited

IEEE 802.15.4 Information Link Type Specification

Version 1

Revision	Description
2019-01-29	Version 1 - Initial Release

February 20, 2019

Contents

1 Introduction

IEEE 802.15.4 is a wireless standard for Low-Rate Wireless Personal Area Networks (LR-WPANs) defining a number of physical (PHY) layers covering a wide variety of freqency bands and a number of Media Access Control (MAC) sub-layers for managing data and management services including beacon management, channel access, frame delivery and validation, and security mechanisms. Developing, debugging, diagnosing, and maintaining technologies using IEEE 802.15.4 require capturing packets with a sniffer. Sniffers generally output captured packets encapsulated in a pcap or pcap-ng packet with a specific data link-type (DLT) [**??**] that is understood by packet analyzers.

Three existing DLTs for IEEE 802.15.4 are defined

- DLT_IEEE802_15_4_WITHFCS (195),
- DLT_IEEE802_15_4_NONASK_PHY (215), and
- DLT_IEEE802_15_4_NOFCS (230).

None of the current DLTs provide a means to include out-of-band meta-data such as received signal strength and channel number which are useful for diagnostics in any wireless transmission system. Also, the current implementation with FCS only supports a 16-bit ITU-T CRC or CRC OK bit in the status bytes provided by TI CC24xx radios, and the FCS type must be selected by the user in the global protocol capture preferences. The IEEE 802.15.4-2015 standard also specifies a 32-bit CRC equivalent to ANSI X3.66-1979 for SUN PHYs or TVWS PHYs. The global protocol configuration does not support configurations with multiple sniffer interfaces each with a different FCS type,

This document proposes and defines a new DLT and format for presenting captured IEEE 802.15.4 packets with meta-data to packet analyzers.

2 Overview

The IEEE 802.15.4 TAP DLT design is based on an extensible Type-Length-Value (TLV) format. The optional meta-data information in encapsulated in one or more TLVs and included in any packet from a sniffer. Some types may be duplicated by pcap-ng options but using the TLVs in this DLT enables providing the meta-data information in both pcap and pcap-ng capture streams. Where types are duplicated by TLVs and pcap-ng options in the same packet, the DLT TLVs have priority.

2.1 PCAPNG Block Types

Although DLTs are common to pcap and pcap-ng, this document will provide examples using pcap-ng only.

2.1.1 PCAPNG IDB

The DLT value is specified in the pcap header or pcap-ng interface description block (IDB) before the the first packet.

```
 0                   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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Block Type = 0x00000001                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Block Total Length                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  DLT_IEEE802_15_4_TAP (283)   |        reserved               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          SnapLen                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Options (variable)                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Block Total Length                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

2.1.2 PCAPNG EPB

An IEEE 802.15.4 TAP packet is encapsulated in the packet data following a pcap record header or in the packet data of a pcap-ng Enhanced Packet Block (EPB).

```
 0                   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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Block Type = 0x00000006                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Block Total Length                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Interface_ID                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Timestamp (High)                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Timestamp (Low)                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Captured Packet Length                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Original Packet Length                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    IEEE 802.15.4 TAP Packet                   |
|                variable length, padded to 32 bits             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Options (variable)                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Block Total Length                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

2.2 IEEE 802.15.4 TAP Packet

The IEEE 802.15.4 TAP Packet consists of the TAP Header, zero or more TLV fields, the PHY payload (PSDU), and optional FCS bytes. All data fields are encoded in little-endian byte order.

2.2.1 IEEE 802.15.4 TAP Header

```
 0                   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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    version    |    padding    |             length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            TAP TLVs                           :
:                        variable length                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              :
:                   IEEE 802.15.4 PHY Payload                   :
:                   [+ FCS per FCS Type TLV]                    :
:                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- version - 0
- padding
- length - total length of header and TLVs in bytes

2.2.2 IEEE 802.15.4 TAP TLVs

IEEE 802.15.4 TAP TLVs have the following format.

```
 0                   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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              type             |             length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             value                             :
:               variable length, padded to 32 bits              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- type - field type identifier
- length - bytes in value field
- value - data for type

3 IEEE 802.15.4 TAP TLV Types

The following subsections describe the currently defined TLVs. Any unknown TLVs should be dissected as a binary blob of the provided length.

3.1 FCS Type

Identifies the FCS type following the PHY Payload. FCS type none (0) is optional if no FCS is present.

```
 0                   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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type = FCS_TYPE (0)      |          datalen = 1          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |              padding                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The FCS type is one of

- 0 = None,
- 1 = 16-bit CRC,
- 2 = 32-bit CRC,
- 3 = TI CC24xx CRC_OK bit.

3.2 Receive Signal Strength

The received signal strength in dBm as a IEEE-754 floating point number.

```
 0                   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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Type = RSS (1)        |          datalen = 4          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          RSS in dBm                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.3 Bit Rate

The transmission data-rate in bps. The bit-rate may change frame to frame in multi-rate PHY configurations.

```
 0                   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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type = BIT_RATE (2)      |           datalen = 4         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Bit rate in bps                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.4 Channel Assignment

Channel assignments are defined through a combination of channel numbers and channel pages. See IEEE 802.15.4-2015 10.1.2 Channel assignments.

```
 0                   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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type = CHANNEL_ASSIGNMENT (3) |           datalen = 3         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Channel number         | Channel page  |    padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.5 SUN PHY Information

An IEEE 802.15.4 SUN PHY is configured to operate in a specified frequency band, encoding type, and rate mode.

```
 0                   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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type = PHY_ENCODING (4)     |         datalen = 3           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Band      |      Type     |      Mode     |   padding     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Band - IEEE 802.15.4 Table 7-19 Frequency band identifer values.
- Type - IEEE 802.15.4 Table 7-20 Modulation scheme encoding values.
- Mode - IEEE 802.15.4 Rate mode depends on the Band and Type.

3.6 Start of Frame Timestamp

The start of frame timestamp is a monotonically increasing time in nanoseconds since power on of the receiving device. This value differs from the timestamp in the pcap or pcap-ng header which is based on the clock time reported by the sniffer.

```
 0                   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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type = SOF_TIMESTAMP (5)    |          datalen = 8          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                          nanoseconds                          |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.7 End of Frame Timestamp

The end of frame timestamp is a monotonically increasing time in nanoseconds since power on of the receiving device. This value is important to time-slotted MACs where a packet may overflow a time slot, or where there are timing constraints on the ack sent in response to a data frame.

```
 0                   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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type = EOF_TIMESTAMP (6)    |          datalen = 8          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                          nanoseconds                          |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.8 Absolute Slot Number (ASN)

For a Time-Slotted Channel Hopping MAC, the Absolute Slot Number (ASN) is a monotonically increasing number which is synchronized across all nodes on the network and forms part of the nonce for decryption.

```
 0                   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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Type = ASN (7)         |          datalen = 8          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                          ASN (64-bits)                        |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3.9 Start of Slot Timestamp

The start of slot timestamp is a monotonically increasing time in nanoseconds since power on of the receiving device. For a Time-Slotted Channel Hopping MAC, the start of slot timestamp, which preceeds the start of frame timestamp, is essential for debugging and optimizing TSCH configurations.

```
 0                   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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type = SLOT_TIMESTAMP (8)   |          datalen = 8          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                          nanoseconds                          |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```


3.10 Timeslot Length

The timeslot length in a Time-Slotted Channel Hopping MAC is used for calculating the delta between end of frame and end of slot.

```
 0                   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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type = TIMESLOT_LENGTH (9)  |          datalen = 8          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Length of timeslot in microseconds            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4 Glossary

Notation	**Description**
DLT	Data-Link Type
EPB	Enhance Packet Block (pcap-ng)
FCS	Frame Check Sequence
IDB	Interface Description Block (pcap-ng)
IEEE	Institute of Electrical and Electronics Engineers
LR-WPAN	Low-Rate Wireless Personal Area Network
MAC	Medium Access Control
pcap	Packet Capture File Format [**??**]
pcap-ng	PCAP Next Generation Capture File Format [**??**]
PDU	Protocol Data Unit
PHY	Physical Layer
SUN	Smart Utility Networks
TLV	Type-Length-Value
TSCH	Time-Slotted Channel Hopping
TVWS	Television White Space

5 References

1. Data-Link Types - `https://www.tcpdump.org/linktypes.html`
2. Packet Capture File Format - `https://wiki.wireshark.org/Development/LibpcapFileFormat`
3. PCAP Next Generation Capture File Format - `https://pcapng.github.io/pcapng/`