03 February 2017 Webex

IPv6 over the TSCH mode of IEEE 802.15.4

Chairs:
Pascal Thubert
Thomas Watteyne

Etherpad for minutes:
http://etherpad.tools.ietf.org:9000/p/6tisch?useMonospaceFont=true

Note Well

This summary is only meant to point you in the right direction, and doesn't have all the nuances. The IETF's IPR Policy is set forth in BCP 79; please read it carefully.

The brief summary:

- By participating with the IETF, you agree to follow IETF processes.
- If you are aware that a contribution of yours (something you write, say, or discuss in any IETF context) is covered by patents or patent applications, you need to disclose that fact.
- You understand that meetings might be recorded, broadcast, and publicly archived.

For further information, talk to a chair, ask an Area Director, or review the following:

- BCP 9 (on the Internet Standards Process)
- BCP 25 (on the Working Group processes)
- BCP 78 (on the IETF Trust)
- BCP 79 (on Intellectual Property Rights in the IETF)

Reminder:

Minutes are taken *
This meeting is recorded **
Presence is logged ***

*   Scribe; please contribute online to the minutes at: http://etherpad.tools.ietf.org:9000/p/6tisch?useMonospaceFont=true
**   Recordings and Minutes are public and may be subject to discovery in the event of litigation.
***   From the Webex login

Agenda

- Administrivia [2min]
  - Agenda bashing
  - Approval minutes from last meeting
- Status of drafts (chairs) [5min]
- News from 802.15.12 (Pat Kinney) [25min]
- Wrap up ML discussions on 6top (cont.) (Thomas) [25min]
- AOB [3min]

Draft status

draft-ietf-6tisch-minimal-19

- IESG telechat
  - Placed on agenda for telechat - 2017-02-16
  - ID Tracker URL: https://datatracker.ietf.org/doc/draft-ietf-6tisch-minimal/
- Published 25 Jan 2017
- Addressing the K1 discussion
- Addressing Ralph’s questions
- Diff at https://tools.ietf.org/rfcdiff?url2=draft-ietf-6tisch-minimal-19.txt

draft-ietf-6tisch-architecture-11

- Published 27 Jan 2017
- Addressing Complex Tracks
- Diff at https://tools.ietf.org/rfcdiff?url2=draft-ietf-6tisch-architecture-11.txt

802.15.12

Project: IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs)

Submission Title: [802.15.12 – Conceptual Overview]
Date Submitted: [31 January 2017]
Source: [Pat Kinney] Company [Kinney Consulting]
Address [Chicago area, IL, USA]
Voice:[+1.847.960.3715], FAX: [Add FAX number], E-Mail:[pat.kinney@kinneyconsultingllc.com]

Re: [Information on IEEE 802.15.12 for IETF coordination effort]

Abstract: [High Level Overview of current state of IEEE 802.15.12]

Purpose: [For informational purposes only]

802.15.12

Conceptual Overview

IEEE 802.15.12 Introduction

- **Introduction**

IEEE 802.15.12 performs many of the functions that an LLC would perform but adds additional functionality needed for IEEE 802.15.4 in areas such as configuration, higher layer protocol identity, fragmentation, harmonization of existing of other upper sublayer layer 2 protocols, and management of 802.15.4 devices.

- **Purpose, to provide the following:**
  - **Reduction of the complexity in configuring and using the 802.15.4 device**
    - **Complexity in configuring 802.15.4** results from having to select one correct configuration given all possible combinations of the following: 8 MAC modes with 13 distinct MAC behaviors, 9 PHY modulation types with 4 distinct PHY behaviors and 40 PHY data rates, and 20 PHY bands with greater than 35,390 channels. 802.15.12 adds a management protocol module to provide configuration parameters to the 802.15.4 device.
    - **Complexity in the use of 802.15.4** to send messages is shown by a comparison with 802.3 and 802.11. Ethernet (802.3) has 4 parameters in its data transmission primitive while 802.11 has 6. However, the 802.15.4 data transmission primitive contains 28 parameters. See Figure 1 for more details. 802.15.12 supplies the additional MCPS parameters over and above 802.3 and 802.11.

IEEE 802.15.12 Introduction

- **Purpose (continued) – provide the following:**
  - **Addition of higher layer protocol identification**
    - An implicit assumption with 802.15.4 is that there is a single application/protocol stack above it, while other standards such as 802.3 and 802.11 use EtherType protocol identities to direct messages to one of many applications. 802.15.12 adds a header supplying higher layer protocol identification.
  - **Fragmentation**
    - While 802.15.4 needs fragmentation due to small frame sizes and low to very low data rates, 802.15.4 does not include frame fragmentation. 802.15.12 provides two fragmentation methods, one for 6LoWPAN operation and the other for all else.
  - **Harmonization**
    - There are numerous layer 2 protocols designed for 802.15.4, however these protocols have not been harmonized to allow combinations of these protocols.
  - **Management**
    - 802.15.4 was not originally intended to be managed, hence the standard did not include managed objects. 802.15.12 introduces managed objects to allow 802.15.4 devices to be managed similar to other devices such as 802.11.

Data Request Comparison - Figure 1

As an example of the complexity of sending or receiving data with 802.15.4 compared to Ethernet or 802.11, the respective data request primitives are shown.

802.3

MA_DATA.request (
destination_address,
source_address,
mac_service_data_unit,
frame_check_sequence
)

802.11

MA-UNITDATA.request (
source address,
destination address,
routing information,
data,
priority,
service class
)

802.15.4

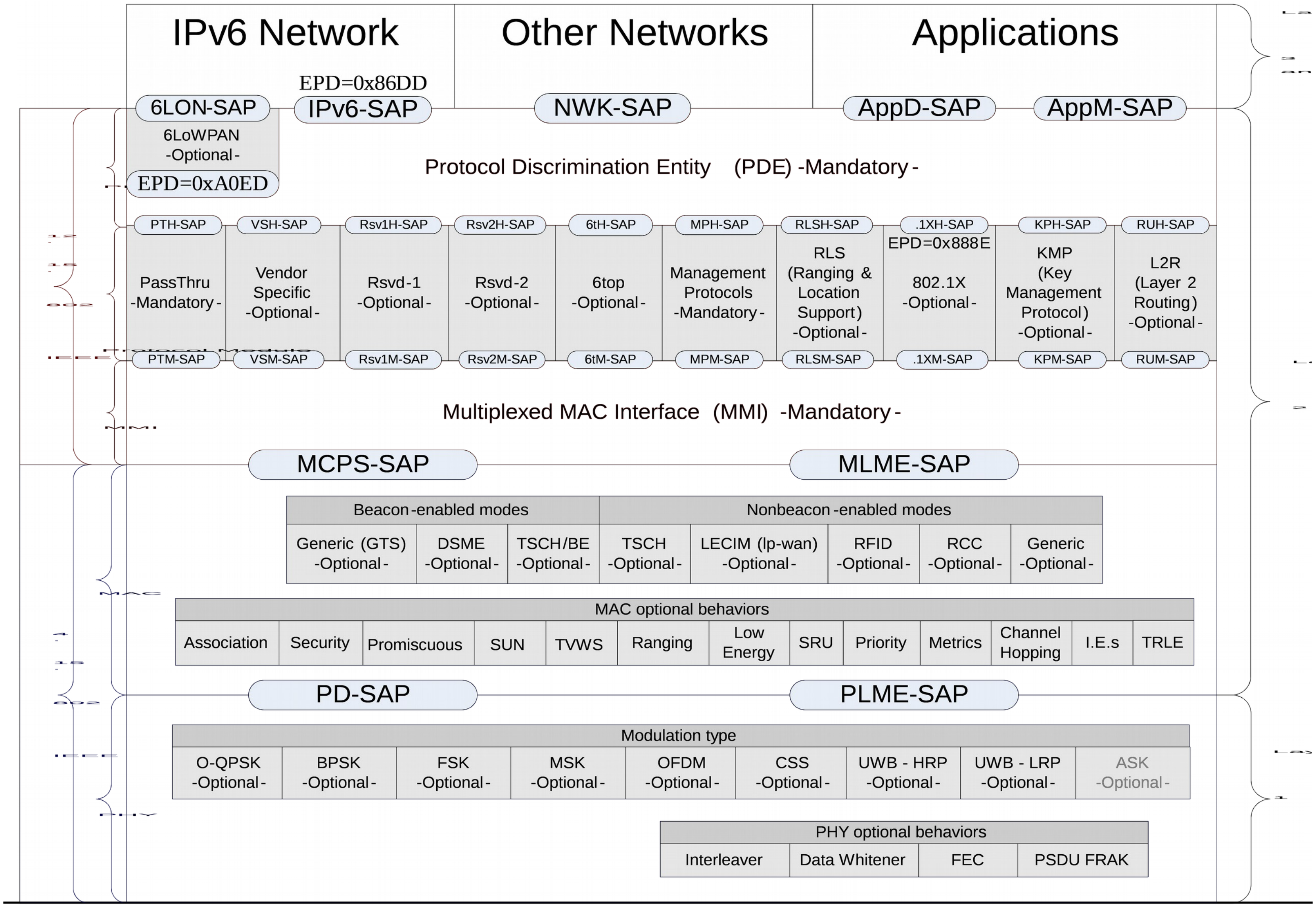
MCPS-DATA.request (
SrcAddrMode,
DstAddrMode,
DstPanId,
DstAddr,
Msdu,
MsduHandle,
HeaderIeList,
PayloadIeList,
HeaderIeIdList,
NestedIeSubIdList,
AckTx,
GtsTx,
IndirectTx,
SecurityLevel,
KeyIdMode,
KeySource,
KeyIndex,
UwbPrf,
Ranging,
UwbPreambleSymbolRepetitions,
DataRate,
LocationEnhancingInformationPostamble,
LocationEnhancingInformationPostambleLength,
PanIdSuppressed,
SeqNumSuppressed,
SendMultipurpose
FrakPolicy,
CriticalEventMessage
)

802.15.12 Functional Decomposition

Overview:

- The 802.15.12 functional decomposition is based upon the 802-2014 Reference Model.
- The functional decomposition as shown in Figure 2 enables:
  - multiple higher layer applications and protocol stacks by use of the Protocol Discrimination Element (PDE). The PDE multiplexes the layer 3 interface to the appropriate protocol module
  - all known Layer 2 protocols for 802.15.4, while still allowing extensibility to add protocols. These protocols are contained within the respective protocol modules. The protocol modules format the layer 3 datagrams into 802.15.4 primitives before transmission and extracts the incoming message from the 802.15.4 primitive for the appropriate layer 3 SAP
  - all protocol modules access to the appropriate 802.15.4 SAP via the Multiplexed MAC Interface (MMI)
  - fragmentation for datagrams, where fragmentation for 6LoWPAN is included in its protocol module and fragmentation for all other messages in included in the Multiplexed MAC Interface (MMI)

PHY and DLL Functional Decomposition Figure 2



802.15.12 Protocol Discrimination Entity (PDE)

Purpose:

•Directs and optionally modifies information from the higher layer SAP to the appropriate protocol module directly or via fragmentation module

•Directs and optionally modifies information from protocol module SAP to the appropriate higher layer SAP directly or via defragmentation module

Overview

•For frames going to the higher layer, the PDE determines the appropriate SAP for delivery, as determined by the ULI header, removes the ULI header, reconstitutes the appropriate header, and then directs the datagram to the SAP.

•For datagrams coming from a higher layer, the PDE determines the SAP to which the datagram is to be sent based upon the configuration of the device as set by the Management Protocols entity, and forwards it to the appropriate SAP.

Further details may be found in 15-16-0656, latest revision

802.15.12 Multiplexed MAC interface (MMI)

Purpose

- Directs and may modify information from a protocol module SAP to the appropriate MAC SAP or another protocol module SAP

Overview

•Provides multiplex and fragmentation service to the packets sent by the ULI functions and send them to either the MCPS-SAP, the MLME-SAP, or to another function module SAP within the ULI. The process of sending the packets includes formatting the ULI IE or prepending the appropriate headers into the payload of the frame for transmission.

•The interface between the MMI and the ULI function modules includes the Multiplex ID and the payload to be sent or the payload received.

•The mechanism for the MMI, i.e. the ability to send the data to the proper SAP, will be similar to the mechanism defined in IEEE 802.15.9 for the multiplexed data service.

Further details may be found in 15-16-0656, latest revision.

802.15.12 Protocol Modules

Purpose:

- Formats messages from the higher layer SAP into the appropriate 802.15.4 primitive requests, e.g. MCPS-DATA.request, for the intended 802.15.4 SAP, or to the appropriate format for the intended protocol module.
- Responds to primitives from an 802.15.4 SAP in an appropriate manner such as sending the MPDU from a MCPS-DATA.indication to the appropriate higher layer SAP, or reacting to a confirm.
- Configures the necessary parameters of the 802.15.4 device for the intended operation such as network operation.

Overview

- The Protocol Module acts as an intelligent interface from the higher layer SAP to the 802.15.4 SAP.
- The Protocol Module works with the PDE and MMI to allow an 802.15.4 device to handle multiple higher level applications.
- There are two mandatory protocol modules: Management Protocol and PassThru.

802.15.12 Mandatory Protocol Modules

Management protocol module provides:

1. Configuration parameters to the MAC and PHY using configuration data received from a higher layer

2. Configuration parameters to other protocol modules received from a higher layer or stored in the management protocol module

Note: ULI Profile IDs, used to identify the device/module configuration, may need to be assigned by the 802.15 ANA for common profiles such as ULI device discovery, etc. However, proprietary configurations will be vendor specific. See 15-17-0050 for more information on ULI Profiles.

- Network device monitoring or management. The monitoring function provides device monitoring metrics to a higher layer application. The management function uses data collected from the device to optimize the device's configuration for better spectral use.
- Discovery services to detect other ULI capable devices.

802.15.12 Mandatory Protocol Modules

PassThru Module has the following functions:

- Allows applications/functions above the ULI to access the 802.15.4 device
- Generates an 802.15.4 primitive for messages from the upper layers as well as the 6LoWPAN protocol module to be passed via the 802.15.4 data SAP (MCPS-SAP)
- Responds to primitives (i.e. MCPS.DATA.confirm and MCPS.DATA.indication) delivered via the data SAP, such as passing the MPDU to a higher layer function

802.15.12 Optional Protocol Modules

- **802.1X** provides authentication, authorization, and cryptographic key agreement mechanisms to support secure communication between end stations connected to 802 networks.
- **802.15.9 (KMP)** provides a methodology to enable key management by providing a transport for key management protocols outside the application layers. Additionally, provides a fragmentation and multiplexing layer for those packets so they can be delivered over smaller MAC layer frames and multiplexed on the recipient end to the right processing service.
- **802.15.10 (L2R**) provides the following functions: topology construction, L2R mesh discovery/join/update/recovery, hop-by-hop retransmission, unicast/multicast/broadcast routing, data concatenation, short address assignment, and security

802.15.12 Optional Protocol Modules

- **6LoWPAN** provides the function of MAC frame modification into a frame format for transmission of IPv6 packets and the method of forming IPv6 link-local addresses and statelessly autoconfigured addresses on IEEE 802.15.4 networks. Additional functions include a header compression scheme using shared context and provisions for packet delivery in IEEE 802.15.4 mesh networks.
- **6tisch** functions as an abstraction of an IP link over the TSCH mode of the MAC sublayer by providing network formation and maintenance, multi-hop topology, assign time source neighbor, resource management, dataflow control, scheduling mechanisms, and security.
- **Ranging and Location Support (RLS):** includes mechanisms for both passive gathering of location enabling information (from the MAC/PHY) and active messaging supporting two-way ranging (and other localization methods), and provides a higher layer application such as a location solver with the location enabling information or with a TOF estimate derived from this.

802.15.12 Device Discovery Techniques

To be able to determine how a message is to be transmitted from the 802.15.4 device, the 802.15.12 ULI will create and populate a table indicating devices that are ULI capable and IE capable.

ULI capable: IE capable

- Reserved for use with devices using 15.4e-2012, or 15.4-2015
  - Payload IE, sent out with defined discovery payload
  - Devices not understanding this IE will reject the IE with no ill effects
  - Devices with 802.15.12 ULI will receive the IE and respond appropriately

ULI capable: IE non-capable

- Reserved for use with devices using older firmware ($\leq$ 2011), i.e. no IEs
- Defined discovery payload is sent using security with a well known ULI key
- Devices not knowing this key will reject packet with no ill effects
- Devices with 802.15.12 ULI will decrypt payload and respond appropriately

802.15.12 Header construction: IE Devices

- ULI IE ID (dedicated to 6LoWPAN traffic)
  - ULI IE ID = total IE length (10 bits), 0b01??, 0b1
  - No Protocol Identifier is required, resulting in a total overhead of 2 octets
- MPX IE (used for all non-6LoWPAN traffic):
  - Defined in 802.15.9, MPX IE ID = total IE length (10 bits), 0b0011, 0b1
  - MPX IE has a length of 2 octets, followed by a transaction control of 1 octet, followed by a Protocol Identifier of 2 octets for a total overhead of 5 octets
  - For the special case where the dispatch code is < 0x001f, the 2-octet Dispatch code is elided, resulting in a total overhead of 3 octets
  - Protocol Identifiers:
- EtherType values are > 0x0600
- Dispatch values assigned by 802.15 ANA are ≤ 0x4FF
- Vendor specific values will be set to 0x565 followed by a 3-octet OUI for that vendor

802.15.12 Header construction: Non-IE devices

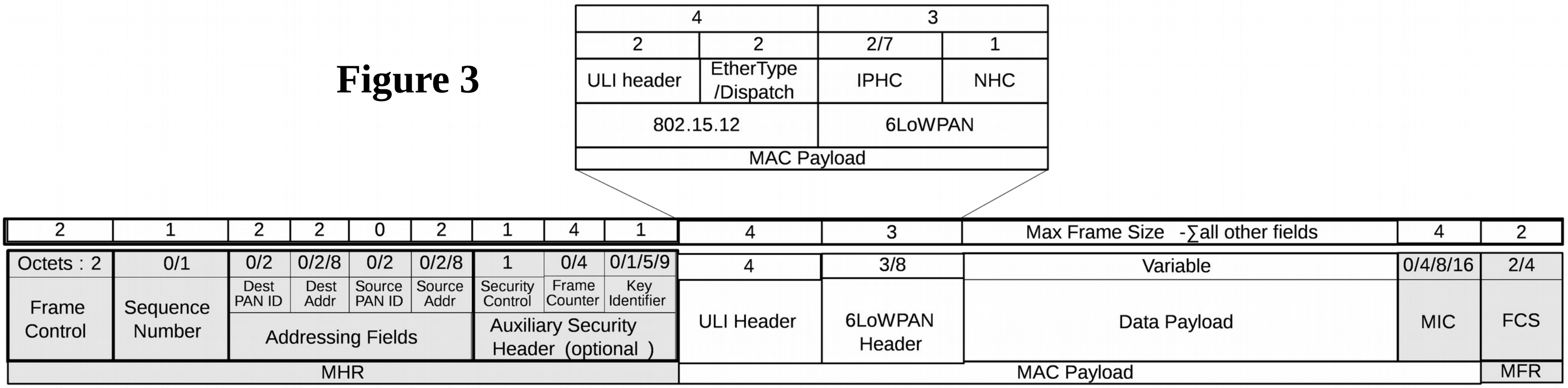
- Non-IE devices
  - 1st payload octet is set to 0xff in accordance with 6LoWPAN Paging Dispatch
  - 2nd payload octet denotes page 15 and will be defined in the future
  - 3rd and 4th payload octets denote the Protocol Identifier
- Note: Non-IE device discovery will use the security mechanism with a "well known" key to effect a discovery ULI packet that will not disturb non-ULI devices. Those 802.15.4 devices not responding to this discovery packet could be assumed to be non-ULI (multiple discovery packets should be sent since a packet may not be received)
- Protocol Identifiers:
  - EtherType values are > 0x0600
  - Dispatch values assigned by 802.15 ANA are ≤ 0x4FF
  - Vendor specific values will be set to 0x565 followed by a 3-octet OUI for that vendor

Examples of Frame Construction with 802.15.12

- The basic assumptions for the following examples of data frames are:
  - 2-octet Frame Check Sequence
  - 2-octet short addresses
  - Origination and Destination devices in same PAN (source PAN ID elided)
  - Security enabled using 4-octet MIC
  - Sequence number is present
  - No header IEs
- Three examples of 802.15.4 data frames using 802.15.12 are shown in the following figures 3 and 4. The examples are:
- Figure 3 - 802.15.4 devices are not IE capable, hence the ULI message is in the payload
- Figure 4 – 802.15.4 devices are IE capable, hence ULI message is in an IE
  - Figure 4a– MPX IE used for all non-6LoWPAN messages
  - Figure 4b– ULI IE used only for 6LoWPAN messages

Frame Composition

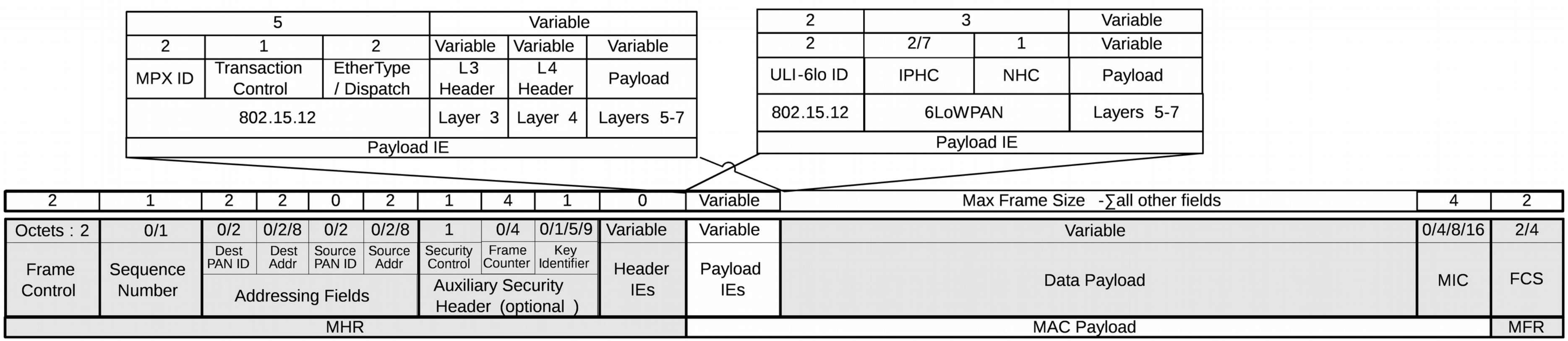
Figure 3



Using Data Payload to convey higher layer data

Figure 4a

Figure 4b

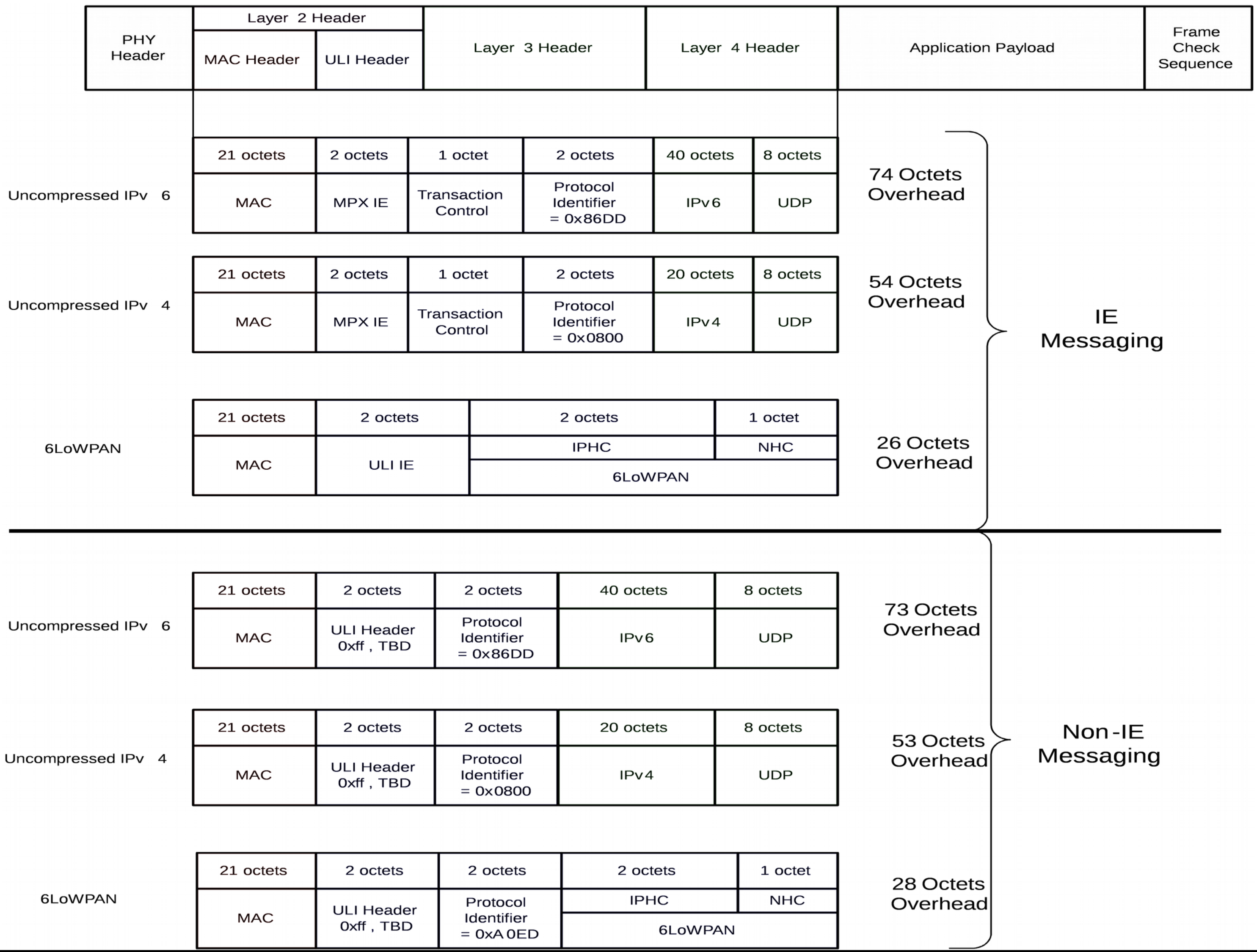


Using IEs to convey higher layer data

Examples of IP Packet Construction using 802.15.12

- Figure 5 shows six examples of IP packets using 802.15.12:
  - IE messaging of non-compressed UDP/IPv6
  - IE messaging of non-compressed UDP/IPv4
  - IE messaging of compressed UDP/IPv6 using 6LoWPAN
  - Non-IE messaging of non-compressed UDP/IPv6
  - Non-IE messaging of non-compressed UDP/IPv4
  - Non-IE messaging of compressed UDP/IPv6 using 6LoWPAN
- All examples use the basic assumptions for frame construction from the previous Frame Construction examples resulting in a 21-octet MAC overhead
- The 6LoWPAN examples are for non-fragmented, no mesh; yielding a 3-octet overhead
- As indicated, the total (MAC + ULI + IP + UDP) header lengths for the six examples range from 26 octets to 74 octets

Packet Construction - Figure 5



Conclusion

Mandatory elements still to be done:

- **PDE**
  - Primitives
  - Parameters
  - Behavior
- **MMI**
  - Primitives
  - Parameters
  - Behavior
- **Management protocol module**
  - Primitives
  - Parameters
  - Behavior
- **Pass-thru protocol module**
  - Primitives
  - Parameters
  - Behavior
- **Networking management**
  - managed objects
  - protocols

Conclusion (continued)

Optional Protocol Modules intended to be done:

- **6LoWPAN**
  - Primitives
  - Parameters
  - Behavior
- **Key Management Protocol (KMP)**
  - Primitives
  - Parameters
  - Behavior
- **Layer 2 Routing (L2R)**
  - Primitives
  - Parameters
  - Behavior
- **6top (layer 2 portion of 6tisch)**
  - Primitives
  - Parameters
  - Behavior
- **Ranging & Location Support (RLS)**
  - Primitives
  - Parameters
  - Behavior

draft-ietf-6tisch-6top-protocol

DONE

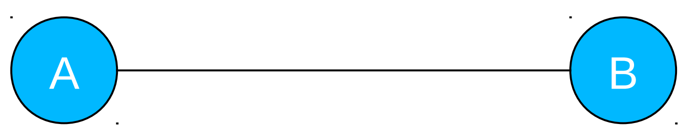
6P signaling traffic

- Is there (should there be) a recommendation on which cells to use for 6P signaling traffic?

DONE

6P RELOCATE command

- In two-step transaction, how to distinguish:
  - Cell which needs to relocated
  - Candidate cells?



DONE

Slotframes & Priorities

6.2.6.4 Multiple slotframes

A given network using timeslot-based access may contain several concurrent slotframes of different sizes. Multiple slotframes may be used to define a different communication schedule for various groups of nodes or to run the entire network at different duty cycles by giving some devices many active timeslots in a slotframe, and others few or none.

A network device may participate in one or more slotframes simultaneously, and not all devices need to participate in all slotframes. By configuring a network device to participate in multiple overlapping slotframes of different sizes, it is possible to establish different communication schedules and connectivity matrices that all work at the same time.

Slotframes can be added, removed, and modified while the network is running. Even though this is the case, all slotframes are aligned to timeslot boundaries, and timeslot 0 of the first repetition of every slotframe is projected back to *macASN* = 0, which is determined by the PAN coordinator (or other network device that starts the network). Because of this, timeslots in different slotframes are always aligned, even though the beginning and end of a particular repetition of that slotframe may not be as illustrated in Figure 6-23. When, for any given timeslot, a device has links in multiple slotframes, transmissions take precedence over receives, and lower *macSlotframeHandle* slotframes takes precedence over higher *macSlotframeHandle* slotframes.

	ASN = 0	ASN = 1	ASN = 2	ASN = 3	ASN = 4	ASN = 5	ASN = 6	ASN = 7	
Slotframe 1 5 slots	TS0	TS1	TS2	TS3	TS4	TS0	TS1	TS2	...
Slotframe 2 3 slots	TS0	TS1	TS2	TS0	TS1	TS2	TS0	TS1	...

Figure 6-23—Multiple slotframes in the network

IEEE Std 802.15.4-2015

3.2. Using 6top with the Minimal 6TiSCH Configuration

6P MAY be used alongside the Minimal 6TiSCH Configuration [I-D.ietf-6tisch-minimal]. In this case, it is RECOMMENDED to use 2 slotframes, as depicted in Figure 3:

- o Slotframe 0 is used for traffic defined in the Minimal 6TiSCH Configuration. In Figure 3, this slotframe is 5 slots long, but it can be of any length.
- o Slotframe 1 is used by 6top to allocate cells from. In Figure 3, this slotframe is 10 slots long, but it can be of any length.

Slotframe 0 SHOULD be of higher priority than Slotframe 1 to avoid for cells in slotframe 1 to "mask" cells in slotframe 0. 6top MAY support further slotframes; how to use more slotframes is out of the scope for this document.

```
                | 0    1    2    3    4  | 0    1    2    3    4  |
                +------------------------+------------------------+
   Slotframe 0  |    |    |    |    |    |    |    |    |    |    |
  5 slots long  | EB |    |    |    |    | EB |    |    |    |    |
 high priority  |    |    |    |    |    |    |    |    |    |    |
                +-------------------------------------------------+

                | 0    1    2    3    4    5    6    7    8    9  |
                +-------------------------------------------------+
   Slotframe 1  |    |    |    |    |    |    |    |    |    |    |
 10 slots long  |    |A->B|    |    |    |    |    |    |B->A|    |
  low priority  |    |    |    |    |    |    |    |    |    |    |
                +-------------------------------------------------+
```

Figure 3: 2-slotframe structure when using 6top alongside the Minimal 6TiSCH Configuration.

draft-ietf-6tisch-6top-protocol

DONE

About Cell Suggestion

```
When in a transaction, node A proposes a candidate CellList to node B
and B cannot allocate any of those cells.  Node B SHOULD respond with
a CellList suggesting alternatives.  This approach facilitates the
agreement between A and B and enables A to not guess what cells may
be not used in B.  The following figure illustrated these 3-step
transaction.

        +----------+                       +----------+
        |  Node A  |                       |  Node B  |
        +----+-----+                       +-----+----+
             |                                   |
             | 6P ADD Request                    |
             |   NumCells     = 2                |
             |   CellList     = [(1,2),(2,2),(3,5)]  |
             |---------------------------------->|
             |                                   |
             | 6P Response                       |
             |   Return Code  = RC_SUCCESS       |
             |   CellList     = [(6,2),(7,2),(8,5)]  |
             |<----------------------------------|
             |                                   |
             | 6P Confirmation                   |
             |   Return Code  = RC_SUCCESS       |
             |   CellList     = [(7,2),(8,5)]    |
             |---------------------------------->|
             |                                   |

       Figure 6: A 3-step 6P Transaction with cell suggestion.
```

Is that what we agreed upon?

DONE?

editorial TODO list

- *All of the above*
- Add RC_END_OF_LIST
- Sentence that says that 3-step transaction used for case where requestee proposes cells
- 3-way → 3-step (same for 2 way)
- Section 4.2.13, the length of "Num. Cells" is 2-octet long in the text, but 1-octet in the figure.
- Figs 5 and 6 need to be re-written to reflect type and code fields
- Single generation number

NumCells in DELETE request

```
4.2.9.  6P DELETE Request Format

                       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| T | R |     Code      |      SFID     | SeqNum|  GEN  |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |          Metadata             |  CellOptions  |   NumCells    |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  | CellList ...
  +-+-+-+-+-+-+-+-+-

[...]

NumCells:  The number of cells from the specified CellList the sender
      wants to delete from the schedule of both sender and receiver.
CellList:  A list of 0, 1 or multiple 6P Cells.  The CellList is an
      opaque set of bytes, sent unmodified to the SF.  The
      RECOMMENDED format of each 6P Cell is defined in Section 4.2.7.
      The SF MAY redefine the format of the CellList field.
```

Should be say that NumCells MUST be equal to the number of cells in the CellList?
Or just remove the NumCells field?

Reordering

renaming

- STATUS -> COUNT?

AOB ?

Thank you!