**IEEE P802.15**

**Wireless Personal Area Networks**

|  |  |
| --- | --- |
| Project | IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs) |
| Title |  |
| Date Submitted | 26th April 2013  |
| Source | René StruikStruik Security ConsultancyToronto, ON Canada | US: +1 (415) 690-7363cell: +1 (647) 867-5658E-mail :  rstruik.ext@gmail.com |
| Re: | This is response to a Call for Contributions for IEEE 802.15.9  |
| Abstract | This document contains guidelines for flagging specific higher-layer protocols and transport support (fragmentation/defragmentation, in-order delivery) as for IEEE 802.15.9 |
| Purpose | This document is intended for inclusion of 802.15.9 draft specification |
| Notice | This document has been prepared to assist the IEEE P802.15. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein. |
| Release | The contributor acknowledges and accepts that this contribution becomes the property of IEEE and may be made publicly available by P802.15. |

***Editorial note: all suggested changes are relative to 80.15.4e-2012. The suggested changes below would allow implementation of information elements of any type that could stretch beyond the maximum size of a MAC MPDU.***

**3.1 Definitions**

***Insert the following new definition:***

**Conceptual object:** octet string that may have syntax or semantics that are only partially specified with 802.15 (e.g., higher-layer information). This conceptual object is only conveyed between devices, after representing this as an ordered sequence of information elements, whereas the receiving entity may only act on this object after converting this back to the corresponding conceptual object. Conceptual objects may be of any size (thus, also of size larger than a MAC MPDU).

***Insert section 5.2.1.7b as indicated:***

**5.2.1.7b Conceptual objects**

**5.2.1.7b.1 General**

A conceptual object is an octet string that may have syntax or semantics that are only partially specified with 802.15 (e.g., higher-layer information). This conceptual object is only conveyed between two devices, after representing this as an ordered sequence of information elements (5.2.1.7b.2.1), whereas the receiving entity may only act on this object after converting this back to the corresponding conceptual object (5.2.1.7b.2.2).

To facilitate description of the mapping from conceptual objects to sequences of information elements (5.2.1.7b.2.1) and the corresponding inverse mapping (5.2.1.7b.2.2), conceptual objects are described as “generalized information elements”, as follows.

Conceptual objects are defined to have a common general format consisting of a 1-octet Conceptual Object Class field and a variable-length object class-specific Information field. Each element is assigned a unique Object class ID as defined in this standard. The format of the Conceptual Object is shown in Fig. 5.2.1.7b.1.

|  |  |  |
| --- | --- | --- |
|  | Conceptual Object Class Id | Conceptual Object Value |
| Octets: | 1 | Variable |
| **Figure 5.2.1.7b.1 – Conceptual object format(11s)** |  |  |

The set of valid object classes is defined in Table 8.4.1a-2.

***Editor note: Conceptual Object Classes 1-3 are based on TLV objects currently defined in P802.15.9/D0.1 (Table 1, p. 6).***

|  |  |  |
| --- | --- | --- |
| **Conceptual Object** | **Conceptual Object Class Id** | **Description** |
| Higher-Layer | 0 |  |
|  | 1-255 | Reserved |

**5.2.1.7b.2 Conversion scheme**

**5.2.1.7b.2.1 Representation of Conceptual Objects**

This involves the transformation of the Conceptual Object field to an ordered sequence of payload information elements, according to the following stipulations:

* Each information element shall be set to the “Encapsulated Service Data Unit IE”, as specified in clause 5.2.4.4;
* The right-concatenation of all IE content in the resulting ordered sequence of information elements shall be equal to the Conceptual Object value;
* The rightmost (last) information element in the resulting ordered sequence of payload information elements shall have its Length field set to zero.
* The representation of an ordered sequence of Conceptual Object fields shall be defined as the ordered sequence of the outcomes of the transformation of each single Conceptual object in this sequence, using this original ordering.

The procedure by which the partitioning of the Conceptual Object field over the Conceptual Object Segment information elements is determined is out of scope of this specification.

**5.2.1.7b.2.2 the Inverse Mapping**

This involves the transformation of an ordered sequence of payload information elements to an ordered sequence of Conceptual Object fields, according to the following stipulations:

* The transformation shall fail if not all payload information elements are “Encapsulated Service Data Unit IE”s, as specified in clause 5.2.4.4;
* The transformation shall partition the ordered sequence of payload information elements into ordered subsequences of information elements, where the rightmost (last) information element in each such ordered subsequence has a Length field set to zero and where each other information element in this subsequence has a Length field set to a nonzero value. If this partitioning is not possible, the transformation shall fail.
* The transformation shall convert each subsequence into a Conceptual Object field which is equal to the right-concatenation of all IE content in the ordered subsequence; the ordered sequence of Conceptual Object fields shall be defined as the result of ordering the Conceptual Object fields reconstructed from each subsequence, according to the partitioning order by which these subsequences were previously derived.

Note: It is easy to see that the composition of these two conversion function results in the original sequence of Conceptual Object fields (if one ignores empty octet strings).

**5.2.1.7b.2.3 Higher-Layer**

The higher-layer conceptual object indicates a “true” higher-layer object. As such, this may be any octet string of length smaller than 0xffff, with syntax and semantics entirely out-of-scope of the standard.

***Editorial note: all suggested changes below are relative to 80.15.9/D01. The suggested changes below would allow implementation of information elements of any type that could stretch beyond the maximum size of a MAC MPDU.***

**5.2.4.2 Header Information Elements**

***Insert the following entry in Table***

|  |  |
| --- | --- |
| ID range | Function |
| 0x18 | Managed ID (described in 802.15.9/D01, 5.3.3.2) |
| 0x19 | Managed ID (described in 5.2.2.4.6a) |

***Editorial note: ID 0x18 is supposed to capture KMPs as currently specified in 802.15.9/D01, clause 5.3.1 (i.e., this refers to key management related transport only that is specifically called out in clause 5.3.3.2). ID 0x019 is supposed to capture \*any\* higher-layer protocols (key establishment, key transport, etc.). With the latter, the specific protocol is to be outsourced to a IANA-style numbering authority. (This would also make sure that with any “new” higher-layer protocol one does not need to start an IEEE process to get a number assigned.)***

***This header information element shall be included with every frame fragment, should fragmentation occur.***

**5.2.2.4.6a “Any” Information Element**

***Insert the following entry in Table x of clause 5.3.3.2:***

|  |  |  |
| --- | --- | --- |
| Higher-Layer Id | Length | Protocol description |
| 0-255 | 0 | *Up to independent IANA-like body (TBD)* |

***Editorial note: These TLV values just specify the type of higher-layer protocol that follows as a sequence of payload information elements (as result of 5.2.1.7b.2.1).***

***Note that one could list the KMP fragments (clause 5.3.3.2 of 802.15.9/D0.1) here….***