IEEE P802.11
Wireless LANs

|  |
| --- |
| LB 200 Resolution to Comments on Traffic Indication Map (TIM) Element  |
| Date: 2013-xx-xx |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| Haiguang Wang  | I2R |  |  | hwang@i2r.a-star.edu.sg  |
| Zander Lei | I2R |  |  | leizd@i2r.a-star.edu.sg |

Abstract

This submission proposes resolutions for the following LB200 comments for sub-clauses 8.4.2.6.1.3 and 8.4.2.6.1.4

CID 1097, 1098, 1099, 1100, 1635, 1636, 2244, 2245, 2246, 2247, 2248

Interpretation of a Motion to Adopt

A motion to approve this submission means that the editing instructions and any changed or added material are actioned in the TGah Draft. This introduction is not part of the adopted material.

***Editing instructions formatted like this are intended to be copied into the TGah Draft (i.e. they are instructions to the 802.11 editor on how to merge the text with the baseline documents).***

***TGah Editor: Editing instructions preceded by “TGah Editor” are instructions to the TGah editor to modify existing material in the TGah draft. As a result of adopting the changes, the TGah editor will execute the instructions rather than copy them to the TGah Draft.***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CID** | **Page** | **Clause** | **Comment** | **Proposed Change** | **Resolution** |
| 1097 | 73.44 | 8.4.2.6.1.4 | "The padding subfield contains 0-7 padding bits. The padding bits also indicate the end of the current ADEBlock." -- how do the padding bits indicate the end of the current block? This is a particularly clever trick when there are 0 of them. | Delete "The padding bits also indicate the end of the current ADE Block." | Revised. Modified the sentence that causing confusion. For clarification, the description on adding padding bits is in the later part of the subclause. TGah editor to make changes shown in 13/1519r0 under the heading for CID 1097. |
| 1098 | 74.04 | 8.4.2.6.1.4 | "If all AIDs in the ADE blocks are paged," -- this language is cellular, not 802.11. No definition exists for an AID being paged.In 802.11 we have bits corresponing to an AID in a virtual bitmap that are equal to 1. | Replace "If all the bits in the virtual bitmap corresponding to the AIDs in the ADE block are equal to 1"Make similar changes to remove all "are paged" language in this subclause. | Rejected Paged AID has been newly defined in 11ah (clause 3.2) |
| 1099 | 74.08 | 8.4.2.6.1.4 | "AP shall set EWL to 7" -- Clause 8 is for structure, not behavior. | If it is necessary to describe an algorithm for encoding in terms of the behavior of the encoder, do so in Clause 9. Otherwise describe the interpretation of the fields here using declarative verbs.Also make similar changes to the 5 other "shall" statements in this subclause. | Agreed in principle. Revised – Change as suggested.TGah editor to make changes shown in 13/1519r0 under the heading for CID 1099. |
| 1100 | 74.27 | 8.4.2.6.1.4 | "For nAID differential values, totally WL +∙ nbits are required." -- that's totally cool, dude | ".. values, a total of WL .." | Revised – Change as suggested.TGah editor to make changes shown in 13/1519r0 under the heading for CID 1100. |
| 1635 | 72.61 | 8.4.2.6.1.3 | OLB is not in the list of acronyms in Section 3.3 | Add OLB to list of acronyms in Section 3.3 | Agreed with the commenter. Revised – Change as suggested. TGah editor to make changes shown in 13/1519r0 under the heading for CID 1635. |
| 1636 | 73.24 | 8.4.2.6.1.4 | ADE is not in the list of acronyms in Section 3.3 | Add ADE to list of acronyms in Section 3.3 | Agreed with the commenter. Revised – Change as suggested.TGah editor to make changes shown in 13/1519r0 under the heading for CID 1636. |
| 2245 | 73.44 | 8.4.2.6.1.4 | There are multiple problems to use padding bits to indicate the end of the ADE block, including"1). what value the padding bits are set to?2). Considering the EWL can be any value between 1 to 8, then the number of padding bits in the last byte could be 1 to 7, then there will be an ambiguity regarding the interpretation of those the last byte, unless the padding bits are set to 0. | Change the paragraph in line 44 to the following:The padding subfield contains 0-7 padding bits. The padding bits are set to 0 and they also indicate the end of the current ADE Block. | Agreed with the commenter. Revised.TGah editor to make changes shown in 13/1519r0 under the heading for CID 2245. |
| 2246 | 74.27 | 8.4.2.6.1.4 | It needs clarification in the sentence in line 12 page 74, regarding the specification of how to encode the Encoded Block Information when only one page AID in ablock, as it uses "May". Then the question is: is there other way to encode than the Signle AID mode? if so, specify it. If not, then change the "May" to "Shall". | Please clarify if there is any other way to encode the case of one page AID with the ADE mode, or just change the "may" to "shall" in the sentence in line 12 page 74. | Agreed in principle that clarificiaton is needed. Revised.TGah editor to make changes shown in 13/1519r0 under the heading for CID 2246. |
| 2247 | 74.27 | 8.4.2.6.1.4 | Where is 31 from in the max number of bits calculation 31\*8=248bits? Is it from the 5-bit length field? If so, this max number of 248bits is misleading, particularly with "shall" in the sentence.Note that if all the page AIDs are in the same Block, the max number of bits is about 160bits. If it allows AIDs in different Blocks in this ADE encoding, then the max number of bits is limited by the Length field. | Clarify the specification about the max number of bits in ADE encoding. | Agreed in principle. Revised.TGah editor to make changes shown in 13/1519r0 under the heading for CID 2247. |
| 2248 | 74.16 | 8.4.2.6.1.4 | Are the AIDs, i.e., from AID1 to AIDn, in the same block?-- If so, why do we need 8 bits to encode the difference of AIDs in the a block with a size of 64 AIDs?-- if not, then it should be clearly specified how the AIDs in multiple Blocks are mapping into the ADE encoding. | Please Clarify. | Agreed in principle. Revised.TGah editor to make changes shown in 13/1519r0 under the heading for CID 2248. |

***TGah editor: insert*** acronyms ***for OLB and ADE in section 3.3 addressing the CIDs 1635 and 1636:***

**3.3 Abbreviations and acronyms**

*OLB Offset Length Block*

*ADE AID with Differential Encoding*

***TGah editor: Modify the 5th paragraph onwards of the sub-clause 8.4.2.1.4 addressing the following CIDs: 1097, 1099, 1100, 2245, 2246, 2247, 2248:***

**8.4.2.6.1.4 ADE mode**

The padding subfield contains 0-7 padding bits. The padding bits are set to 0 ~~also indicate the end of the current ADE Block.~~

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Bits | 3 | 5 | 1-8 |  | 1-8 | 0-7 |
|  | EWL | Length | *ΔAID1* | … | *ΔAIDn* | Padding |
| Octets |  1 |  |  | 0-31 |  |  |

To encode a list of paged AIDs, denoted as AID1, AID2 … AIDn, an AP can derive the offset value in the Block Offset field (8.4.2.6.1) for the current ADE Block by $\left⌊(AID1 modulo 2048)/64\right⌋$, where $\left⌊x\right⌋ $refers to the largest integer that is not larger than *x*.

The encoding procedure is as follows:

If all AIDs in the ADE blocks are paged, AP sets the Inverse Bitmap subfield to 1 and ADE Block consists only EWL and Length fields, where both EWL and Length Field are set to zeros.

If all but one AIDs in the ADE blocks are paged, AP sets the Inverse Bitmap subfield to 1 and ADE Block consists only one ΔAID subfield. AP sets EWL to 7 and Length subfield to one. ΔAID subfield is set to (AID – (Page Index × 2048 + Block Offset × 64)).

If only one AID is paged in the ADE blocks, AP sets the Inverse Bitmap subfield to 0 and uses the Single AID mode.

For all other cases, AP sorts all AID*i*, *i* = 1, 2~~,~~…~~,~~ *n* in an ascending order (AID1 < AID2 < … < AID*n*) and then calculates the AID differential values according to:

ΔAID1 = AID1 – (Page Index × 2048 + Block Offset × 64)

ΔAID*i* = AID*i* – AID*i*-1, *i* = 2~~,~~ …~~,~~ *n*.

Determine WL as the minimum bits that can represent the largest ΔAID*i* (or mathematically $\left⌊log\_{2}^{MAX(ΔAIDi)}\right⌋+1$ where MAX(ΔAID*i*) denotes the largest ΔAID*i*), *i*=1,2,…,*n*. The value of EWL subfield is set to WL-1. For *n* AID differential values, a total of WL × *n* bits are required. The number of bits ~~shall be~~ is less than or equal to 248 since maximum payload in an ADE Block is 31×8=248 (31 is the maximum value in the 5-bit Length subfield). If ~~T~~the total number of bits WL × *n* ~~may~~is not ~~be~~ a multiple of an octet~~.~~, zero bits are padded to make the ADE Block end at octet boundary.

When decoding, if Inverse Bitmap subfield is 1, EWL and Length subfield are zeros, all AIDs in the ADE blocks are paged.

If Inverse Bitmap subfield is 1, EWL is 7 and Length subfield is 1, all AIDs except one in the ADE blocks are paged. The unpaged AID is ΔAID1 + Block Offset × 64 + Page Index×2048.

For other cases, a STA extracts Page Index (8.4.2.6) and Block Offset (8.4.2.6.1), EWL and Length values from the respective fields. It derives WL by adding 1 to the value from EWL field. The paged AIDs are then derived with following formulas:

AID1 = ΔAID1 + (Page Index × 2048 + Block Offset × 64)

AID*i* = ΔAID*i* + AID*i* - 1, *i* = 2~~,~~ …~~,~~ *n*.

The decoder stops the decoding of the current ADE block when either one of following conditions is satisfied:

1. T~~t~~he number of bits left for decoding is less than WL

1. ΔAIDi is zero and *i*>1.

AIDs from the same or different blocks can be encoded into one ADE Block. A STA can derive the number of AIDs, including both paged and unpaged AIDs, encoded in one ADE Block with following method:

If an ADE Block is not the last encoded Block in the TIM element, the decoder can derive the number of AIDs encoded by this ADE Block based on the Block offset values in the current and the immediate next encoded blocks. For example, the offset values in the current ADE Block and the next encoded Block are Offset1 and Offset2. Then the AIDs encoded by this ADE Block is [Page Index × 2048 + Offset1×64, Page Index × 2048 + Offset2×64), Page Index×2048 + Offset1×64 is included and Page Index×2048 + Offset2×64 is excluded.

If an ADE Block is the last one in the TIM element, the number of AIDs encoded by the last ADE Block can be determined based on the offset value and Page length or segment length if its TIM Page is segmented.