IEEE P802.11
Wireless LANs

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|

|  |
| --- |
| CR on TXTIME and PSDU\_LENGTH |
| Date: 2017-03-14 |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| Youhan Kim | Qualcomm |  |  | youhank@qca.qualcomm.com |
|  |  |  |  |  |

 |

Abstract

This submission proposes resolutions for the following comments from the letter ballot on P802.11ax D1.0:

3441, 9490, 8566

NOTE – Set the Track Changes Viewing Option in the MS Word to “All Markup” to clearly see the proposed text edits.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CID** | **Commenter** | **Page** | **Clause** | **Comment** | **Proposed Change** |
| 3441 | Albert Petrick | 371.36 | 28.4.2 | Clarify "aSignalExtension" referenced from Table 19-25 | Add the following underlined text (without the underline):and is aSignalExtension (0us for 5 GHz band, 6us for 2.4 GHz band) |

**Discussion**

Corresponding text from D1.1 is the following (P381):

|  |
| --- |
|  |

From IEEE802.11-2016 P2426, aSignalExtension is defined as:

|  |
| --- |
|  |

The commenter is suggesting to reiterate within Clause 28 that aSignalExtension is 0 usec in the 5 GHz band, and 6 usec in the 2.4 GHz band. However, D1.1 clearly specifies that aSignalExtension is “*as defined in Table 19-25*”, where it is unambiguous that the aSignalExtension takes the values as proposed by the commenter.

However, there does seem to be an issue in referencing the TXVECTOR parameter NO\_SIG\_EXTN because the NO\_SIG\_EXTN parameter is not present for HE PPDUs according to D1.1 P225:

|  |
| --- |
|  |

As all HE PPDUs must “include” Signal Extension (including the case of adding 0 usec of Signal Extension in case of 5 GHz band), there is no need to reference the non-existent NO\_SIG\_EXTN parameter. I.e., a more proper definition of THE\_PREAMBLE would be:

|  |
| --- |
|  is defined as in Equation (28-116) and Equation (28-117), and *SignalExtension* takes the value of aSignalExtension as defined in Table 19-25 (HT PHY characteristics)  |

**Proposed Resolution: CID 3441**

**Revised**. Table 19-25 has the information the commenter is requesting to add. Rather than duplicating the information, it would be better to refer to Table 19-25. It is also noted that the reference to the TXVECTOR parameter NO\_SIG\_EXTN is incorrect as the parameter is not included for HE PPDUs (see Table 28-1). Hence, the proposed resolution updates the draft text to remove this error and clearly refer readers to Table 19-25 for the definition of aSignalExtension.

TGax editor: Replace the D1.1 P381L36-39 with “T\_{HE\_PREAMBLE} is defined as in Equation (28-116) and Equation (28-117), and *SignalExtension* takes the value of aSignalExtension as defined in Table 19-25”.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CID** | **Commenter** | **Page** | **Clause** | **Comment** | **Proposed Change** |
| 9490 | Yan Zhang | 371.58 | 28.4.2 | "TPE is given by Equation (28-113)." quoted wrong equation. It should be Equation (28-117). | Replace (28-113) with (28-117) |

**Discussion:**

Corresponding text in D1.1 is the following (P381):

|  |
| --- |
| … |

where Equation (28-113) is the following (P347):

|  |
| --- |
|  |

This is applicable only for HE MU PPDU, thus the commenter is correct that the reference needs to be updated.

Note that the commenter is suggesting to replace the reference to Equation (28-117).

D1.1 P348:

|  |
| --- |
|  |

But Equation (28-117) is for the receiver side, thus is not the appropriate reference for TXTIME computation at the transmitter side. Rather, the PE\_DURATION field in the TXVECTOR should be used in computing the TXTIME.

D1.1 P234:

|  |
| --- |
|  |

Hence, the sentence under question should be updated as:

|  |
| --- |
| *TPE* is given by the TXVECTOR parameter PE\_DURATION. |

**Proposed Resolution: CID 9490**

**Revised.** The commenter is correct that Equation (28-113) is not the appropriate reference for T\_PE. However, Equation (28-117) is also not the appropriate reference as it is for computation at the RX side, while the comment was on the TXTIME computation at the TX side. Rather, the PE\_DURATION field in the TXVECTOR should be used.

TGax editor: Change “Equation (28-113)” on D1.1 P381L58 to “the TXVECTOR parameter PE\_DURATION”.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CID** | **Commenter** | **Page** | **Clause** | **Comment** | **Proposed Change** |
| 8566 | ron porat | 372.00 | 28.4.2 | In an 11ac MU transmission with mixed BCC/LDPC users, the airtime for all the users was the same. However, the PSDU length as defined in 28-131 of 11ax D1.0 does not satisfy this property. In particular, for a mixture of BCC and LDPC users, when an LDPC extra symbol is added, BCC users' PSDU will not align with the rest | Ensure same airtime for all users, irrespective of coding. Update equation 28-131 as follows:a. Use two separate equations, one for the BCC and one for the LDPC case.b. Equation for LDPC case: same as equation 28-131, except that we omit the '-- N\_tail' at the endc. Equation for BCC case: same as equation 28-131, except that we use Nsym instead of Nsym,init, and NDBPS,last,u instead of NDBPS,last,init,u |

**Discussion:**

Following is the relevant text from D1.1 (P382):

|  |
| --- |
|  |

To illustrate the error in Equation (28-131), let us walk through the process of transmission up to the computation of PSDU\_LENGTH.

1. MAC uses the PLME-TXTIME.request primitive (IEEE802.11-2016 P614) to inform PHY how much payload MAC wishes to transmit per user (APEP\_LENGTH*u*).
2. PHY computes *Nsym,init* and *ainit*, which is common across all users (D1.1 P327).

|  |
| --- |
| … |

1. Based on *ainit*, PHY computes the initial number of (uncoded) data bits which can fit in the last OFDM symbol for each user.

|  |
| --- |
|  |

1. By default,

|  |
| --- |
|  |

But if any user employing LDPC requires extra LDPC symbol segment, then *a* is “incremented by 1”

|  |
| --- |
|  |

1. The number of *coded* bits which is transmited in the last OFDM symbol is then

|  |
| --- |
|   |

1. At this point, note that the coded bits (*NCBPS,last,u*) covers one more “symbol segment” than the uncoded bits (*NDBPS,last,init,u*) if an extra LDPC symbol segment was required in step #4 above.
	1. In case of LDPC users, the LDPC encoding scheme generates additional coded bits (by repeating coded bits if necessary) to ‘fill’ the extra symbol segment. Thus, there is no need to ‘increase’ the number of *uncoded* data bits to fill the extra symbol segment.
	2. In case of BCC users, however, there is no functionality within the BCC encoder which generates extra coded bits to fill the extra symbol segment. Hence, the number of *uncoded* data bits has to be increased to cover the extra symbol segment as well. This is done on D1.1 P329:

|  |
| --- |
|  |

1. MAC needs to be told of the ultimate PSDU size it is allowed to transmit (including any extra space created due to the extra symbol segment). This is the PSDU\_LENGTH returned from PHY to MAC via the PLME-TXTIME.confirm primitive. Unfortunately, Equation (28-131) does not reflect the potential increase in PSDU\_LENGTH for BCC users due to the extral symbol segment. This is the error the commenter is pointing out. The fix for this is mostly inline with the proposed resolution by the commenter.

During this review, however, more issues have been found in the TX encoding parameter computation and encoding process. For example, suppose MAC told PHY that it wants to transmit APEP\_LENGTH = 5,000 bytes for a particular user in an HE MU PPDU to PHY via the PLME-TXTIME.request primitive. After the BCC/LDPC parameter computations, PHY replies back saying that the PSDU\_LENGTH for the user is 5,500 bytes. What is the MAC supposed to do w/ the extra 500 bytes it now has? That is clearly answered in D1.1 subclause 27.10.2 and IEEE802.11-2016 subclase 10.13.6.

D1.1 P199:

|  |
| --- |
|  |

IEEE802.11-2016 P1369-1370:

|  |
| --- |
| … |

The yellow highlighted part indicates that even though the MAC initially intended to transmit 5,000 bytes, it is allowed to add additional A-MPDUs given that it now has 500 extra bytes available for its use. Finally, if the MAC no longer has any more A-MPDUs to add, then the MAC needs to add EOF padding subframe/octets to pad the remaining PSDU\_LENGTH.

The issue in D1.0/1.1 is that PHY subclause also describes about what MAC is supposed to do with the extra 500 bytes in the PSDU\_LENGTH. For example, D1.1 P329 writes:

|  |
| --- |
|  |

However, PHY actually does not know whether MAC is ‘padding’ the difference between the APEP\_LENGTH and PSDU\_LENGTH, or if MAC is transmitting additional A-MPDUs. To call potential A-MPDUs as pre-FEC padding bits is misleading. From the perspective of PHY, once the PSDU\_LENGTH has been computed, PHY should no longer care about APEP\_LENGTH. It is MAC’s duty to deliever a PSDU of PSDU\_LENGTH, regardless of what the actual content of that PSDU is. PHY should simply encode and transmit the PSDU.

While the discussion in this document has focus on HE MU PPDU, similar issue is present for HE SU PPDU descriptions as well. The proposed text updates in this document addresses both cases.

Another issue is that the PSDU\_LENGTH compute at the transmitter is used to populate the RXVECTOR.

D1.1 P382:

|  |
| --- |
|  |

In 11ac, the PSDU\_LENGTH computation for the RXVECTOR was described in the 21.3.20 PHY Receive Procedure subclause, but no such description is available in the TGax D1.1. Hence, the proposed resolution adds text to derive PSDU\_LENGTH at the receiver.

Note, however, that PSDU\_LENGTH computation for the RXVECTOR when receiving HE trigger based PPDU has not been addressed in the proposed resolution because there seems to be a more fundamental question of how does receiver is ‘told’ on the various parameter required for reception. E.g. how is the PHY told of the MCS, Nss, PSDU\_LENGTH, etc? Do we need to define a new PLME? Also, as the PHY is “somehow” told of all these parameters, does the PHY need to include them again in the RXVECTOR? This seems to be major topic on its own, deserving a separate contribution.

**Proposed Resolution: CID 8566**

**Revised**. The commenter is correct that Equation (28-131) is erroneous in computing the PSDU\_LENGTH. While updating the draft text to address the issue, various other issues pertaining to the PSDU\_LENGTH was found in the draft. The proposed resolution addresses all these issues, except for the PSDU\_LENGTH computation when receiving an HE trigger based PPDU, which seems to require a separate contribution.

TGax editor: Implment the text changes under the “Proposed Text Updates: CID 8566” section in 11-17/0465r0.

**Proposed Text Updates: CID 8566**

*TGax Editor: Update D1.1 P324L42 as shown below:*

For an HE SU PPDU or an HE extended range SU PPDU, PSDU\_LENGTH is then computed using Equation (28-130). This PSDU\_LENGTH ensures that MAC delivers a PSDU that fills the available octets in the Data field of the HE PPDU upto the boundary represented by the *ainit* value in the last OFDM symbol(s). The PHY then determines the number of remaining pad bits to add using Equation (28-63) and appends them to the PSDU. Note that this number of pre-FEC pad bits added by PHY (*NPAD,Pre-FEC*) will always be 0 to 7.

  (28-63)

*TGax Editor: Add equation number to the equation at D1.1 P329L20 as shown below:*

For the users with BCC encoding, update the *NDBPS* of the last symbol as



 (28-86a)

*TGax Editor: Update D1.1 P329L33 as shown below:*

PSDU\_LENGTH for each user is then computed using Equation (28-131) for users employing BCC, and using (28-131a) for users employing LDPC encoding. The PSDU\_LENGTH ensures that MAC delivers a PSDU for each user that fills the available octets in the Data field of the HE PPDU upto the boundary in the last OFDM symbol(s) represented by *ainit* for users encoded by LDPC, and represented by *a* for users encoded by BCC.  PHY then determines the number of remaining pad bits to add for each user using Equation (28-87) for users employing BCC encoding, and Equation (28-87a) for users employing LDPC encoding, and appends the pad bits to the PSDU. Note that this number of pre-FEC pad bits added by PHY (*NPAD,Pre-FEC,u*) will always be 0 to 7.

  (28-87)

 (28-87a)

For each user with either LDPC or BCC encoding, the number of post-FEC padding bits in each of the last *mSTBC* symbol(s) is computed as in Equation (28-88):

  (28-88)

*TGax Editor: Add the following text at D1.1 P380L65:*

For HE SU or HE extended range SU PPDUs, the value of the PSDU\_LENGTH parameter returned in the RXVECTOR is calculated using Equation (28-128a).

  (28-128a)

where *NSYM,RX* is given by Equation (28-128b)

 *mSTBC* is 1 if the STBC field in HE-SIG-A is 0, and 2 if the STBC field is 1

 *NDBPS,last,RX* is given by Equation (28-128c)

 *R* is the code rate

 *NDBPS* is defined in Table 28-12

 *Nservice, Ntail* are defined in Table 28-9

  (28-128b)

where *NSYM* is given by Equation (28-116)

  (28-128c)

where *aRX* is given by Equation (28-128d)

 *NSD,SHORT* is defined in Table 28-25

 *NSS, NBPSCS* are defined in Table 28-12

  (28-128d)

where *a* is the value of the Pre-FEC Padding Factor field in HE-SIG-A

For HE MU PPDUs, the value of the PSDU\_LENGTH parameter for user *u* returned in the RXVECTOR is calculated using Equation (28-128e).

  (28-128e)

where *NSYM,RX* is given by Equation (28-128f)

 *mSTBC* is 1 if the STBC field in HE-SIG-A is 0, and 2 if the STBC field is 1

 *NDBPS,last,RX,u* is given by Equation (28-128g)

 *R* is the code rate

 *NDBPS,u* is defined in Table 28-12

 *Nservice, Ntail* are defined in Table 28-9

  (28-128f)

where *NSYM* is given by Equation (28-116)

  (28-128g)

where *aRX* is given by Equation (28-128h)

 *NSD,SHORT,u* is the *NSD,SHORT* defined in Table 28-25 for user *u*

 *NSS,u, NBPSCS,u* are defined in Table 28-12

  (28-128h)

where *a* is the value of the Pre-FEC Padding Factor field in HE-SIG-A

*TGax Editor: Update D1.1 P382L13 as shown below:*

The value of the PSDU\_LENGTH parameter for user *u* returned in the PLME-TXTIME.confirm primitive for an HE MU PPDU is calculated using Equation (28-131) and Equation (28-131a) for users using BCC and LDPC encoding, respectively.

 (28-131)

 (28-131a)

where

*NSYM,init* is given by Equation (28-78)

*NDBPS,last,u* is given by Equation (28-86a)

*NDBPS,last,init,u* is given by Equation (28-79)

[End of File]