IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Resolutions for DCF and EDCA comments on 11md/D0.1 | | | | |
| Date: 2017-07 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Graham SMITH | SR Technologies | Davie, FL, USA. | 916 799 9563 | gsmith@srtrl.com |

Abstract

This submission proposes resolutions for CIDs 294,282,255,200,227,365,364

Green indicates material agreed to in the group,

yellow material to be discussed, red material rejected by the group and

cyan material not to be overlooked.

The “Final” view should be selected in Word.

R3 Re-worked CIDs 294, 189, 282

R4-7 reworking CID 282

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CID | Commenter | Clause | Page Line | Comment | Proposed |
| 294 | Mark RISON | 10.3.3 |  | Is the backoff timer in units of slots or in units of time (multiples of aSlotTime)? For example, Equation (10-1) indicates the backoff time (or Backoff Time) is in units of time, but at 1351.61 the backoff timer is "decremented" and 1351.8 says the backoff timer has a value measured in "backoff slots" | Be consistent. It's probably easiest if the backoff timer is something that counts in units of slots |
| 189 |  | 10.22.2.4 | 1486.27 | “backoff timer” is confusing since it is not really a timer, it’s just a counter (and indeed there are a number of “backoff counters) | Change all instances of “backoff timer” in the draft to “Backoff counter” |

Discussion:

Equation 10.1 is clear Backoff Time = Random()  aSlotTime

As aSlotTime is in µs then Backoff Time is in µs.

“After this DIFS or EIFS medium idle time, the STA shall then generate a random backoff period (defined by Equation (10-1)) for an additional deferral time before transmitting,”

Let’s look at the defining paragraphs for the DCF backoff procedure.

1429.22 Clause 10.3.4.3

“If no medium activity is indicated for the duration of a particular backoff slot, then the backoff procedure shall decrement its backoff timer by aSlotTime.

If the medium is determined to be busy at any time during a backoff slot, then the backoff procedure is suspended; that is, the backoff timer shall not decrement for that slot. The medium shall be determined to be idle for the duration of a DIFS or EIFS, as appropriate (see 10.3.2.3 (IFS)), before the backoff procedure is allowed to resume. Transmission shall commence when the backoff timer reaches 0.”

The major question (and I argued this in 11mc) is what does ‘resume’ mean? Does it mean:

1. Having to wait a full aSlotTime, i.e. starting the slot again, or,
2. Resuming at the point during the slot where the timer stopped?

A) implies that the backoff timer is in multiples of aSlotTime, whereas B) implies microseconds.

If we assume A) is correct then a STA could be stuck in a timeslot, everytime waiting DIFS. I admit that as written A) is a more likely interpretation than B), and also the EDCA version is clearly all in timeslot increments. Also in EDCA version the timer is definitely in units of timeslots. Hence, although I still think this is inefficient, **we have to conclude that the backoff timer (counter) is in units of timeslots.**

Propose that this is made clear by an insertion at 1398.64.

As to “backoff timer vs backoff counter, I agree with the commenter, it is a counter.

After input from Mark H, the question comes down, I think, to what to do with Equation 10-1

1. Keep as is. Then it is a time in microseconds
2. Change to be a counter then Backoff Counter = Random ( )

We choose Backoff Counter = Random ( ) which had coinsensus at first review.

**RESOLUION**

**REVISED, make changes a shown**

1398.63

After deferral, or prior to attempting to transmit again immediately after a successful transmission, the STA shall select a random backoff count (see equation 10.1) and shall decrement the backoff counter once per interval of aSlotTime while the medium is idle.

1426.41

After this DIFS or EIFS medium idle time, the STA shall then generate a random backoff count (defined by Equation (10-1)) for an additional deferral time before transmitting unless the backoff counter already contains a nonzero value, in which case the selection of a random number is not needed and not performed.

1426.48

Backoff Count = Random()

1429.20

A STA performing the backoff procedure shall use the CS mechanism (see 10.3.2.1 (CS mechanism)) to determine whether there is activity during each backoff slot. If no medium activity is indicated for the duration of a particular backoff slot, then the backoff procedure shall decrement its backoff timer by aSlotTime.

If the medium is determined to be busy at any time during a backoff slot, then the backoff count shall not decrement for that slot. The medium shall be determined to be idle for the duration of a DIFS or EIFS, as appropriate (see 10.3.2.3 (IFS)), plus aSlotTime before the backoff count is allowed to decrement for that slot.

Transmission shall commence when the backoff counter equals 0.”

*(Note to editor: This sentence is now a seperate para)*

1429.41

If the transmission is successful, the CW value reverts to aCWmin before the random backoff count is chosen, and the SSRC and/or SLRC are updated as described in 10.3.3 (Random backoff time). The result of this procedure is that transmitted frames from a STA are always separated by at least DIFS.

1429.57

Within an IBSS a separate backoff count shall be generated to precede the transmission of a Beacon frame

At 1486.28

Each EDCAF shall maintain a backoff counter, which has a value measured in backoff slots as described below

When the backoff procedure is invoked, the backoff counter is set to an integer

At 1487.19

- Decrement the backoff ~~timer~~counter

1487.31

At each of the above-described specific slot boundaries, each EDCAF shall decrement the backoff counter if the

backoff counter for that EDCAF has a nonzero value.

1487.37 and 1487.47

The backoff counter for that EDCAF has a value of 0, and

At 1489.40

… and the backoff ~~timer~~counter has a value of 0.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CID | Commenter | Clause | Line | Comment | Proposed |
| 282 |  | 10.3.4.4 |  | There are several issues with the [QS]SRC/LRC stuff: sometimes it's per-MPDU (p.1290 in mc/D6.0), sometimes it's per-MSDU/MMPDU (p.1295); sometimes it's Data frames only (p.1290), sometimes Management too (p.1295) | Frankly, I can't work it out. Tell me whether SRC/LRC is per-MPDU or per-MSDU/MMPDU, and whether it includes Managament frames/MMPDUs, and I'll come up with the changes |

Discussion:

SRC = short retry count LRC = Long retry count

SSRC= Station short retry count SLRC = station long retry count

dot11ShortRetryLimit

This attribute indicates the maximum number of transmission attempts of a frame, in a PSDU of length that is less than or equal to dot11RTSThreshold, that are made before a failure condition is indicated

dot11LongRetryLimit

This attribute indicates the maximum number of transmission attempts of a frame, in a PSDU of length that is greater than dot11RTSThreshold, that is made before a failure condition is indicated."

So

SRC and SSRC are concerned with frames, PSDUs, that are less than the RTS Threshold and therefore do not require the use of RTS/CTS.

And

LRC and SLRC are concerned with frames, PSDUs, that are greater than the RTS Threshold limit and hence do require RTS/CTS.

The basic intention is that a SSRC/SLRC is for all transmitted packets, whereas SRC and LRC are on a specific MPDU/MMPDUs. As we only have one short and one long “dot11 limit” so this applies to both.

1426.60

“Every STA shall maintain a STA short retry count (SSRC) as well as a STA long retry count (SLRC), both of which shall take an initial value of 0. The SSRC shall be incremented when any short retry count (SRC) associated with any MPDU with the Type subfield equal to Data is incremented. The SLRC shall be incremented when any long retry count (LRC) associated with any MPDU with the Type subfield equal to Data is incremented.”

So SSRC and SLRC only increment with SRC and LRC when the packet is a data packet? But…

1431.41

“A STA shall maintain a SRC and an LRC for each MSDU or MMPDU awaiting transmission. These counts are incremented and reset independently of each other.”

SO… what else do we read

1431.51

“After transmitting a frame that requires acknowledgment, the STA shall perform the acknowledgment procedure, as defined in 10.3.2.9 (Acknowledgment procedure). The SRC for an MPDU with the Type subfield equal to Data or Management and of length less than or equal to dot11RTSThreshold and the SSRC shall be incremented every time transmission of that MPDU fails.”

**This clearly says SRC and SSRC incremented for a specific data or management frame.**

We then read 1431.45

***“After an RTS frame is transmitted, the STA shall perform the CTS procedure, as defined in 10.3.2.7 (CTS and DMG CTS procedure). If the RTS frame transmission fails, the SRC for the MSDU or MMPDU and the SSRC are incremented. This process shall continue until the number of attempts to transmit that MSDU or MMPDU reaches dot11ShortRetryLimit.***

HANG ON – RTS is sent for LONG PACKETS, so what MSDU or MMPDU? This is incorrect. Also the last sentence implies only for **that** MSDU or MMPDU which is a LONG PACKET. This then surely is the LSR?

As Adrian pointed out, RTS is a short packet so it is a short packet that failed – but it is not a data or management packet, it is a control packet so can’t relate to SRC.

One last observation – if a short packet fails, then inevidably a long packet would have failed, so it makes sense to increment the SLRC if a short packet fails.

What this scheme should have been:

* The counts, SRC and SSRC are for Data and Management frames (and RTS).
* SRC/LRC is incremented on a specific failed Data or Management short/long frame (no RTS/RTS))
* SSRC/SLRC is incremented for all failed Data or Management short/long frames (no RTS/RTS).
* Failure of RTS/CTS counts as failed short frame
* Failed short frames should also increment the SLRC

Having analysed this as much as possible we come back to Adrian’s standard bar:

1. Do not break anything
2. Legacy cannot be disadvantaged
3. Performance not damaged

The PROBLEM is the section on RTS. RTS is a control frame and it clearly states that SRC etc. are incremented for data and management frames. Also this section refers to “If the RTS frame transmission fails, the SRC for the MSDU or MMPDU and the SSRC are incremented.” What MSDU or MMPDU??? A reasonable interpretation of this is the MSDU or MMPDU that was associated with the RTS. But this is a Long Packet, so it can only be the LSC and SLRC that can be incremented. SRC is related to a particular MSDU/MMPDU and RTS is neither, so which SRC is incremented?? Clearly can’t do it and this is broken.

Hence, I think we have we have three options for this RTS section (cannot leave it as is):

OPTION 1 – Delete it

OPTION 2 – Change SRC and SSRC to LRC and SLRC

“After an RTS frame is transmitted, the STA shall perform the CTS procedure, as defined in 10.3.2.7 (CTS and DMG CTS procedure). If the RTS frame transmission fails, the LRC for the MSDU or MMPDU and the SLRC are incremented. This process shall continue until the number of attempts to transmit that MSDU or MMPDU reaches dot11LongRetryLimit

OPTION 3 - Delete references to MSDU and MMPDU, and only increment SSRC if RTS/CTS fails.

“After an RTS frame is transmitted, the STA shall perform the CTS procedure, as defined in 10.3.2.7 (CTS and DMG CTS procedure). If the RTS frame transmission fails, the SSRC is incremented.

Below I have chosen Option #2 as this has the least changes.

Proposed resolution:

REVISED

In Clause 10.3.3

Make changes as follows from 1426.60 “Every STA shall maintain a STA short retry count (SSRC) as well as a STA long retry count (SLRC), both of which shall take an initial value of 0. The SSRC shall be incremented when any short retry count (SRC) associated with any MPDU with the Type subfield equal to Data or Management is incremented. The SLRC shall be incremented when any long retry count (LRC) associated with any MPDU with the Type subfield equal to Data or Management is incremented. The CW shall take the next value in the series every time an unsuccessful attempt to transmit an MPDU causes either STA retry counter to increment, until the CW reaches the value of aCWmax. A retry is defined as the entire sequence of frames sent, separated by SIFSs, in an attempt to deliver an MPDU, as described in Annex G. Once it reaches aCWmax, the CW shall remain at the value of aCWmax until the CW is reset. This improves the stability of the access protocol under high-load conditions. See Figure 10-13 (Example of exponential increase of CW).

The CW shall be reset to aCWmin after every successful attempt to transmit a frame containing all or part of an MSDU or MMPDU, when SLRC reaches dot11LongRetryLimit, or when SSRC reaches dot11ShortRetryLimit.

The SSRC shall be reset to 0 when a CTS frame is received in response to an RTS frame, when a BlockAck frame is received in response to a BlockAckReq frame, when an Ack frame is received in response to the transmission of a frame containing all or part of an MSDU or MMPDU that is contained in a PSDU of length less than or equal to dot11RTSThreshold, or when a frame with a group address in the Address 1 field is transmitted.

The SLRC shall be reset to 0 when an Ack frame is received in response to transmission of a frame containing all or part of an MSDU or MMPDU that is contained in a PSDU of length greater than dot11RTSThreshold, or when a frame with a group address in the Address 1 field is transmitted.”

In Clause 10.3.4.4 make changes as follows:

At 1431.45

“After an RTS frame is transmitted, the STA shall perform the CTS procedure, as defined in 10.3.2.7 (CTS and DMG CTS procedure). If the RTS frame transmission fails, the LRC for the MSDU or MMPDU and the SLRC are incremented. This process shall continue until the number of attempts to transmit that MSDU or MMPDU reaches dot11LongRetryLimit.

After transmitting a frame that requires acknowledgment, the STA shall perform the acknowledgment procedure, as defined in 10.3.2.9 (Acknowledgment procedure). The SRC for an MPDU with the Type subfield equal to Data or Management and of length less than or equal to dot11RTSThreshold and the SSRC shall be incremented every time transmission of that MPDU fails. This SRC and the SSRC shall be reset when transmission of that MPDU succeeds. The LRC for an MPDU with the Type subfield equal to Data or Management and of length greater than dot11RTSThreshold and the SLRC shall be incremented every time transmission of that MPDU fails. This LRC and the SLRC shall be reset when transmission of that MPDU succeeds. All retransmission attempts for an MPDU with the Type subfield equal to Data or Management that has failed the acknowledgment procedure one or more times shall be made with the Retry subfield set to 1.

Retries for failed transmission attempts shall continue until the SRC for the MPDU with the Type subfield equal to Data or Management is equal to dot11ShortRetryLimit or until the LRC for the MPDU with the Type subfield equal to Data or Management is equal to dot11LongRetryLimit. When either of these limits is reached, retry attempts shall cease, and the MPDU with the Type subfield Data (and any MSDU of which it is a part) or Management shall be discarded. A DMG STA, in addition to using random access within a CBAP, may transmit retries in available scheduled SPs.”

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CID | Commenter | Clause | Line | Comment | Proposed |
| 255 | Mark Rison | 10.22.2.4 |  | "At each of the above-described specific slot boundaries, each EDCAF shall decrement the backoff timer if the  backoff timer for that EDCAF has a nonzero value.  At each of the above-described specific slot boundaries, each EDCAF shall initiate a transmission sequence if  [...]  --- The backoff timer for that EDCAF has a value of 0, and  [...]  At each of the above-described specific slot boundaries, each EDCAF shall report an internal collision (which  is handled in 10.22.2.4 (Obtaining an EDCA TXOP)) if  [...]  --- The backoff timer for that EDCAF has a value of 0, and" -- this could be read as saying that if the backoff timer is 1 at the slot boundary, you decremement it, and then transmit/internally collide (because it is now 0) | Make it clear that you don't do more than one of the actions. You either decrement, or you transmit, or you internally collide, or you do nothing. Adding "Otherwise" to the beginning of the non-first "At each of"s would be better than nothing |

Discussion:

I would bring attention to 16/0228 where similar issues were discussed.

10.22.2.1 P1483.28

*“****The EDCA channel access protocol is derived from the DCF procedures*** *described in 10.3 (DCF) by adding four independent enhanced distributed channel access functions (EDCAFs) to provide differentiated priorities to transmitted traffic, through the use of four different access categories (ACs).”*

Hence, one would expect that EDCA procedures are similar to DCF but with AIFS[AC] replacing DIFS.

At 1486.45

*EDCAF operations shall be performed at slot boundaries, defined as follows on the primary channel, for each EDCAF:*

This is followed by a list a) to f). There is a problem with f)

*f) Following aSlotTime of idle medium, which occurs immediately after any of these conditions, a) to f), is met for the EDCAF.*

It refers to itself which is wrong, it should read a) to e).

Here is the section cited:

P1487.16

*On these specific slot boundaries, each EDCAF shall make a determination to perform one and only one of the following functions:*

*— Decrement the backoff timer.*

*— Initiate the transmission of a frame exchange sequence.*

*— Invoke the backoff procedure due to an internal collision.*

*— Do nothing.*

*NOTE—If an EDCAF gains access to the channel and transmits MSDUs, A-MSDUs, or MMPDUs from a secondary*

*AC, the EDCAF of the secondary AC is not affected by this operation. If the EDCAF of a secondary AC experiences an*

*internal collision with the EDCAF that gained access to the channel, it performs the backoff procedure regardless of the*

*transmission of any of its MSDUs, A-MSDUs, or MMPDUs (see 10.22.2.6 (Sharing an EDCA TXOP)).*

Three of the four ‘performances’ are then each further described, and this may be the cause of the confusion as cited by the commenter as it appears that they are additional rather than expansions. The position of the “note” does not help either as it seperates the two related sections.

*At each of the above-described specific slot boundaries, each EDCAF shall decrement the backoff timer if the*

*backoff timer for that EDCAF has a nonzero value.*

*At each of the above-described specific slot boundaries, each EDCAF shall initiate a transmission sequence if*

*— There is a frame available for transmission at that EDCAF, and*

*— The backoff timer for that EDCAF has a value of 0, and*

*— Initiation of a transmission sequence is not allowed to commence at this time for an EDCAF of higher UP.*

*At each of the above-described specific slot boundaries, each EDCAF shall report an internal collision (which*

*is handled in 10.22.2.4 (Obtaining an EDCA TXOP)) if*

*— There is a frame available for transmission at that EDCAF, and*

*— The backoff timer for that EDCAF has a value of 0, and*

*— Initiation of a transmission sequence is allowed to commence at this time for an EDCAF of higher UP.*

I do not see that the “one and only one” need be stated, either the rule is clear or it is not and there should not be a choice.

Poll

1. Changes similar to proposed 5
2. Move “Note” 9
3. Reject 7

**Personally, I consider the proposed text to be clearer. We have had many comments on this text and this proposed text would satisfy most if not all of them. I urge others to look again. There is no change in behaviour being proposed.**

Proposed resolution:

REVISED

Make changes as shown:

At P1487.13

f) Following aSlotTime of idle medium, which occurs immediately after any of these conditions, a) to e), is met for the EDCAF.

At P1487.16

At each of the above-described specific slot boundaries:

Either each EDCAF shall decrement the backoff counter if the backoff counter for that EDCAF has a nonzero value.

Or each EDCAF shall initiate a transmission sequence if

— There is a frame available for transmission at that EDCAF, and

— The backoff counter for that EDCAF has a value of 0, and

— Initiation of a transmission sequence is not allowed to commence at this time for an EDCAF of higher UP.

Or each EDCAF shall report an internal collision (which is handled in 10.22.2.4 (Obtaining an EDCA TXOP)) if

— There is a frame available for transmission at that EDCAF, and

— The backoff timer for that EDCAF has a value of 0, and

— Initiation of a transmission sequence is allowed to commence at this time for an EDCAF of higher UP.

Otherwise do nothing.

NOTE—If an EDCAF gains access to the channel and transmits MSDUs, A-MSDUs, or MMPDUs from a secondary

AC, the EDCAF of the secondary AC is not affected by this operation. If the EDCAF of a secondary AC experiences an

internal collision with the EDCAF that gained access to the channel, it performs the backoff procedure regardless of the

transmission of any of its MSDUs, A-MSDUs, or MMPDUs (see 10.22.2.6 (Sharing an EDCA TXOP)).

OPTIONAL RESOLUTION

REVISED

In response to comment, the text “perform one and only one” is deemed to be clear.

Move “Note” at 1487.25 to 1487.51

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CID | Commenter | Clause | Line | Comment | Proposed |
| 200 | Mark Rison | 10.22.2.4 |  | What exactly is "the backoff procedure" for EDCA? Is it the thing which starts with waiting for AIFS of idleness, or the thing which starts with throwing a random number and then waiting for that number of slots of idleness, or what? The term is used in many places, but this is hard if it's not defined! Note that for DCF it's well-defined: "To begin the backoff procedure, the STA shall set its backoff timer to a random backoff time using the equation  in 10.3.3 (Random backoff time)." | At the start of the second para of 10.22.2.4 change "When the backoff procedure is invoked" to "To begin the backoff procedure" |

Discussion:

P1483.51

10.22.2.2. EDCA backoff procedure

Back off procedure 10.22.2.2 only describes when it is invoked, it does not actually describe the back off procedure at all.

P1485.14

*The backoff procedure shall be invoked by an EDCAF when any of the following events occurs*

P1485.61

*If the backoff procedure is invoked for reason c), d), e), or f) above*

Hence we could consider if this clause should be “Invocation of EDCA backoff procedure” or simply “EDCA backoff”?

P1486.25

10.22.2.4 Obtaining an EDCA TXOP

*Each EDCAFshall maintain a backoff timer, which has a value measured in backoff slots as described below.*

*When the backoff procedure is invoked, the backoff timer is set to an integer value chosen randomly with a uniform distribution taking values in the range [0,CW[AC]] inclusive.*

Discussion: Would need to go back and look at each use of “invoke”.

Proposed resolution:

REJECT

The phrase “Invoke the backoff procedure” is in 6 places. The text at 1486.30 makes it clear what this means.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CID | Commenter | Clause | Line | Comment | Proposed |
| 227 | Mark Rison | 10.22.2.7 | 1492.6 | It says " the TXOP  holder shall set the TXVECTOR parameter CH\_BANDWIDTH of a PPDU as follows". It is not clear whether the TXOP holder is required to do this for all PPDUs it transmits, or just for one of them | Delete "of a PPDU" in the cited text at the referenced location |

Discussion:

Note correct Clause, Page and line.

**10.22.2.7 Multiple frame transmission in an EDCA TXOP**

*If a TXOP is protected by an RTS or CTS frame carried in a non-HT or a non-HT duplicate PPDU, the TXOP holder shall set the TXVECTOR parameter CH\_BANDWIDTH of a PPDU as follows:*

*— To be the same or narrower than RXVECTOR parameter CH\_BANDWIDTH\_IN\_NON\_HT of the last received CTS frame in the same TXOP, if the RTS frame with a bandwidth signaling TA and TXVECTOR parameter YN\_BANDWIDTH\_IN\_NON\_HT set to Dynamic has been sent by the TXOP holder in the last RTS/CTS exchange.*

*— Otherwise, to be the same or narrower than the TXVECTOR parameter CH\_BANDWIDTH of the*

*RTS frame that has been sent by the TXOP holder in the last RTS/CTS exchange in the same TXOP.*

In a TXOP there may be multiple frame transmissions, so more than one PPDU. So the question is whether the TXOP holder sets every PPDU or any PPDU. I would assume the intention is that every PPDU in the TXOP is so set. So alternatives might be:

*If a TXOP is protected by an RTS or CTS frame carried in a non-HT or a non-HT duplicate PPDU, the TXOP holder shall set the TXVECTOR parameter CH\_BANDWIDTH of each PPDU as follows:*

*If a TXOP is protected by an RTS or CTS frame carried in a non-HT or a non-HT duplicate PPDU, the TXOP holder shall set the TXVECTOR parameter CH\_BANDWIDTH as follows:*

So we will go with the proposed.

Discussion , in the Std. in this concept, the term “a PPDU” is interpreted as “every PPDU”.

Proposed resolution:

REJECT

In the Std., in this context, the term “a PPDU” is interpreted as “every PPDU” and therefore the instruction is unambiguous.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CID | Commenter | Clause | Line | Comment | Proposed |
| 365 | Woojin Ahn | 10.22.2.4 | 1487.50 | an EDCAF may reach a slot boundary where the backoff timer has a value of 0 without having any pending frame (e.g., expiration of MSDU lifetime/ after transmission as a non-TXOP holder). There's no normative behavior for such condition. We think thath it is better to initialize an EDCAF when its bacff counter reaches zero and there's no available frame. | Add the following.  At each of the above-described specific slot boundaries, each EDCAF shall initialize EDCAF  -- There is NO frame available for transmission at that EDCAF, and  -- The backoff timer for that EDCAF has a value of 0 |

Discussion:

To start the EDCA procedure there must be a frame queued for transmission (see P1485.16).

I cannot find anything in the EDCA section on what happens if an MSDU lifetime expires. If this means that there is no frame in the transmit queue, then one would suppose that the EDCA is cancelled.

In DCF and EDCA immediately after having successfully transmitted a packet the STA backs off using CWmin, and then counts down.

So, if an MSDU lifetime is reached (due to excessive back off?), should the STA:

1. still count down from where it is, wait for timer = 0 and then do not immediately back
2. assume the condition that a frame has been transmitted, and immediately backs off using CWmin
3. assume the condition that there is no frame to be transmitted and the backoff has reached 0.

The commenter is proposing a)

In an implementation I am not sure if the backoff procedure is aware that the MSDU has expired, a) sort of presumes it may not. However, if a STA is suffering badly and timesout its MSDU, or has chosen a very high CW, is it fair it must stick with it? If it has another MSDU queued up should it be able to throw the dice again straightway or still wait?

I would lean towards b).

Proposed resolution:

REJECT

If no frame is available, then the normative behavior is the “Do nothing” case at 1487.24.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CID | Commenter | Clause | Line | Comment | Proposed |
| 364 | Woojin Ahn | 10.22.2.2 | 1486.8 | When an AP updates EDCA parameter set, associated STAs only update their MIB attributes and their current CW[AC] will remain the same. Therefore, when a STA enters the "otherwise" condition, it might have CW[AC] less than CWmin[AC] or greater than CWmax[AC] if its EDCA parameter set had been updated by the associated AP. It is necessary to keep STAs' CW values within the intended range [CWmin, CWmax]. | - Otherwise,  - If CW[AC] is less than CWmax[AC], CW[AC] shall be set to the value max(CWmin[AC], (CW[AC] + 1) ├ù 2 - 1).  - If CW[AC] is equal to or greater than CWmax[AC], CW[AC] shall be set to CWmax[AC]. |

Discussion:

Text in question is

1486.8

*— Otherwise,*

*— If CW[AC] is less than CWmax[AC], CW[AC] shall be set to the value (CW[AC] + 1) × 2 – 1.*

*— If CW[AC] is equal to CWmax[AC], CW[AC] shall be left unchanged.*

Proposal is to change the second bullet to account for CW[AC] > CWmax[AC]

Proposed resolution:

REVISED

1486.8 to read

— Otherwise,

— If CW[AC] is less than CWmax[AC], CW[AC] shall be set to the value (CW[AC] + 1) × 2 – 1.

— Else CW[AC] shall be set to CWmax[AC].