IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CID 1505 | | | | |
| Date: August 20, 2018 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Menzo Wentink | Qualcomm | Utrecht, The Netherlands | +31-65-183-6231 | mwentink  @qualcomm.com |

Abstract

This document proposes a resolution for CID 1505.

**Comment**

|  |  |  |
| --- | --- | --- |
| CID 1505  Mark Rison  10.3.3 | 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**

**EDCA**

The current EDCA text related to retry counters is cited below, dissected and reordered. Retry counters and the associated rules are marked bold.

**10.23.2.12 Retransmit procedures**

**10.23.2.12.1 General**

A QoS STA shall maintain a **short retry counter** and a **long retry counter** for each MSDU, A-MSDU, or MMPDU that belongs to a TC that requires acknowledgment. The initial value for the **short and long retry counters shall be 0**.

The **short retry count** for an MSDU or A-MSDU that is not part of a block ack agreement or for an MMPDU **shall be incremented** every time transmission of a frame in a PSDU of length **less than or equal to dot11RTSThreshold** fails for that MSDU, A-MSDU, or MMPDU.

QoS STAs also maintain a short retry counter and a long retry counter for each AC. They are defined as **QSRC[AC]** and **QLRC[AC]**, respectively, and **each is initialized to a value of 0**.

[The following are the rules for the short retry count and QSRC]

After an RTS frame is transmitted to protect an MSDU or MMPDU, a QoS STA performs the CTS procedure, as defined in 10.3.2.8 (CTS and DMG CTS procedure).

If a valid CTS frame is received, the **QSRC[AC]** for the corresponding AC **shall be reset to 0**.

This **short retry count** and theQoS STA **QSRC[AC] shall be reset** when an A-MPDU or frame of length in a PSDU less than or equal to dot11RTSThreshold succeeds.

If a valid CTS frame is not received, the **short retry counter** for the MSDU or MMPDU and the **QSRC[AC]** for the corresponding AC **shall be incremented**.

**QSRC[AC] shall be incremented** every time transmission of a frame in a PSDU of length less than or equal to dot11RTSThreshold fails, regardless of the presence or value of the DEI field.

For internal collisions occurring with the EDCA access method, the appropriate retry counters (**short retry counter** for MSDU, A-MSDU, or MMPDU and **QSRC[AC]** or long retry counter for MSDU, A-MSDU, or MMPDU and **QLRC[AC]**) **are incremented**.

[The following are the rules for the long retry count and QLRC]

The **long retry count** for an MSDU or A-MSDU that is not part of a block ack agreement or for an MMPDU **shall be incremented** every time transmission of a MAC frame in a PSDU of length **greater than dot11RTSThreshold** fails for that MSDU, A-MSDU, or MMPDU.

This **long retry count** and the **QLRC[AC] shall be reset** when an A-MPDU or frame in a PSDU of length greater than dot11RTSThreshold succeeds.

**QLRC[AC] shall be incremented** every time transmission of an A-MPDU or frame in a PSDU of length greater than dot11RTSThreshold fails, regardless of the presence or value of the DEI field.

For internal collisions occurring with the EDCA access method, the appropriate retry counters (**short retry counter** for MSDU, A-MSDU, or MMPDU and **QSRC[AC]** or long retry counter for MSDU, A-MSDU, or MMPDU and **QLRC[AC]**) **are incremented**.

**10.23.2.2 EDCA backoff procedure**

If the backoff procedure is invoked for reason c), d), e), or f) above, or the transmission failure of a non-initial frame by the TXOP holder, the value of CW[AC] shall be updated as follows before invoking the backoff procedure:

— If the **QSRC[AC]** or the **QLRC[AC]** **has reached** **dot11ShortRetryLimit** or **dot11LongRetryLimit** respectively, CW[AC] shall be reset to CWmin[AC].

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].

**10.23.2.12 Retransmit procedures**

**10.23.2.12.1 General**

Retries for failed transmission attempts shall continue until one or more of the following conditions occurs:

— The **short retry count** for the MSDU, A-MSDU, or MMPDU is equal to **dot11ShortRetryLimit**.

— The **long retry count** for the MSDU, A-MSDU, or MMPDU is equal to **dot11LongRetryLimit**.

When any of these limits is reached, retry attempts shall cease, and the MSDU, A-MSDU, or MMPDU shall be discarded.

Some observations on these rules:

The short retry count is updated for short packets, like short MSDUs and RTS.

The long retry count is updated for long packets, which are MSDUs longer than dot11RTSThreshold.

A STA maintains retry counts per packet (short retry count / long retry count) and for the STA overall (QSRC/QLRQ).

For a long packet, RTSs are transmitted until a CTS is received. The short retry count and QSRC are updated during that process. When a CTS is received but the long packet fails, the long retry count and QLRC are updated.

Therefore, the long retry count and QLRC increase only when an RTS/CTS exchange already succeeded and a collision was not detected. So it seems strange that the long retry count (QLRC) ties into the backoff procedure for resetting the CW.

There appears to be little use for having two retry counters.

Also, the "has reached" language suggests that the CW keeps resetting once a retry count is equal to or larger than its limit. In an alternative interpretation, "has reached" means "is equal to", and the CW resets only once. This is kind of unclear and probably neither behavior was the intended one:

— If the QSRC[AC] or the QLRC[AC] has reached dot11ShortRetryLimit or dot11LongRetryLimit respectively, CW[AC] shall be reset to CWmin[AC].

A possible way forward might be to replace the dual retry counter structure with a single backoff stage that governs the CW and a single retry count that governs packet discard.

Legacy implementations with the dual retry counter structure will still be compliant under these simplified rules, also because the retry limits are configurable by a STA and an AP, so there is not one single behavior described by the current rules.

**10.23.2.2 EDCA backoff procedure**

If the backoff procedure is invoked for reason c), d), e), or f) above, the value of CW[AC] shall be updated as follows before invoking the backoff procedure:

Associated with each AC are

* a Backoff Stage BS[AC], which controls the size of CW[AC]
* a configurable maximum backoff stage BSmax[AC]), which shall be larger than 12

For each AC (omitting [AC] for each parameter):

— If BS is less than BSmax,

* BS shall be increased by 1
* CW shall be set to the lesser of CWmax and 2^BS × (CWmin + 1) - 1

— Else (BS equals BSmax)

* BS shall be set to 0
* CW shall be set to CWmin

Any text related to QSRC and QLRC is deleted (deletions are not shown here).

A single retry count remains, which determines when a packet must be discarded:

A QoS STA shall maintain a retry counter for each MSDU, A-MSDU, or MMPDU that belongs to a TC that requires acknowledgment. The initial value for the retry counter shall be 0.

The retry count for an MSDU or A-MSDU that is not part of a block ack agreement or for an MMPDU shall be incremented every time transmission fails for that MSDU, A-MSDU, or MMPDU, including of an associated RTS.

Retries for failed transmission attempts shall continue until one or more of the following conditions occurs:

— The retry count for the MSDU, A-MSDU, or MMPDU is equal to dot11RetryLimit.

When this limit is reached, retry attempts shall cease, and the MSDU, A-MSDU, or MMPDU shall be discarded.

**Proposed Resolution**

TBD.