IEEE P802.11
Wireless LANs

|  |
| --- |
| Proposed resolution to CID 2462 |
| Date: 2014-05-14 |
| Author(s): |
| Name | Company | Address | Phone | email |
| Mark Hamilton | Spectralink | 2560 55th StBoulder, CO 80301 USA | +1.303.441.7553 | mark.hamilton@spectralink.com |

Abstract

This submission proposes a resolution to REVmc CID 2462.

The discussion is in reference to Draft P802.11REVmc\_D2.8.

R0 – Initial draft

R1 – edited per comments at face-to-face session

This document presents a proposed resolution to REVmc CID #2462.

| **CID** | **Page** | **Clause** | **Comment** | **Proposed Change** | **Owning Ad-hoc** |
| --- | --- | --- | --- | --- | --- |
| 2462 | 1417.42 | 10.3.5.4 | Reassociation to the same AP behaviour is not clearly defined (e.g. effect on TSes, whether failure leaves you unassociated, whether you need to re-do 4WH, meaning of PM bit in Reassociation Request, etc.) | Clarify that reassociation to the same AP transitions the non-AP STA to not be in power save mode. No other behaviors are affected. | MAC |

**Discussion:**

4.5.3.4 states:

Reassociation also enables changing association attributes of an established association while the STA remains associated with the same AP.

So, some special behavior is anticipated, otherwise the STA could simply do a new (full) Association. The question is what behavior differences are there from a (full) Association, or from a Reassociation to a different AP.

Thus, we consider the various state variables, state information and agreements that exist and are shared between the AP and non-AP STAs, and the effect on each of a Reassociation to the same AP:

Obviously, the link/association state is as described in 10.3, and reassociation does affect this per those rules.

Security experts have confirmed that a new PTKSA (and GTKSA and IGTKSA) needs to be established after a reassociation, even a reassociation to the same AP (assuming not FT – can one FT to the same AP?). This is already present in the current text:

10.3.5.4: “Except when the association is part of a fast BSS transition, the SME shall delete any PTKSA and temporal keys held for communication with the AP or PCP(11ad) by using the MLME-DELETEKEYS.request primitive”

The 802.1X controlled port state is reset, of course, to match the PTKSA reset. Likewise, Supplicant state machinery all reset – but only for the link to the AP (not for other security sessions, if any).

Specification and testing by a ‘major certification organization’ defines (and explicitly tests) that any Traffic Streams in place at the time of a reassociation to the same AP will remain in effect after the reassociation. If the reassociation fails, the non-AP STA is assumed to be disassociated and all TSes are, of course, deleted then.

10.4.9.1 says:

All TSPECs that have been set up shall be deleted upon disassociation and reassociation. Reassociation causes the non-PCP and non-AP STA and HC, DMG AP, or PCP to clear their state, and the non-PCP and non-AP STA has to reinitiate the setup.

This doesn’t unambiguously cover the reassociation to the same AP case, but since it doesn’t say otherwise, it must be assumed to cover that case. Thus, the current text is not aligned with the certification testing.

The discussion of reassociation to the same AP is all moot for mesh STAs, which are covered by 13.3 peering procedure instead of (re)association.

Similarly, IBSS has no association (and presumably, no reassociation, therefore), and so is also moot.

What about PBSS? Is Reassociation with PCP allowed? Nothing seems to say it isn’t. So, the proposal is to treat this case the same as AP reassociation.

An exhausted[sic] search leads to these other “state variables”, “state information”, “sessions”, “relationships”, “allocations”, and “agreements” to consider (open to discussion – the treatment of these is somewhat arbitrary and subjective, but this seems like a reasonable set):

Propose to **reset/re-initialize/delete**:
- EDCAF state (largely moot, since a frame exchange just completed, but to be clear, any TXOP also ends)
- Various types of Block Ack agreements
- Sequence number(s) *– implies duplicate cache*
- PN counter(s)

- Flush all transmit packets waiting in queues
- Power Management mode (PM bit should be ignored on Reassociation frame – part of another comment)
 *Already covered in current text, in 10.2.2.2:* A non-AP STA shall be in Active mode upon Association or Reassociation.(#79)- WNM Sleep mode

Propose to **maintain the state** across the reassociation:
- PSMP sessions (because TSes are maintained)
- Enablement/Deenablement (10.12.2) – why not?
- GDD enablement – why not?
- STSL, DLS agreements, TDLS agreements, SMKSA, STKSA, TPKSA – kept, with all peers, which aren’t aware of the reassociation, anyway. However, 10.7.6 says: “The STKSA remains valid even if the STA disassociates from the originating AP, but the STKSA shall be deleted before a STA attempts another association or reassociation.” So, clarify that there is an exception for reassociation to the same AP

- MMSL – kept, with all peers, (some of) which aren’t aware of the reassociation, anyway
- GCR agreement – why not?
- DMS agreement. Note that this procedure is already (almost) explicit, in 10.24.16.2:
 A DMS recipient(#2050) may request modification of the traffic characteristics or attributes of one or more accepted DMS traffic flows by sending a DMS Request frame or Reassociation Request frame containing one or more DMS Descriptors with the Request Type set to “Change” … *(And, similar for the “Delete” case.)*
- TFS agreement – why not?
- FMS agreement – why not?
- Triggered autonomous reporting (of measurements). This is already covered explicitly in 10.11.8:
 A STA in an infrastructure BSS or PBSS(11ad) shall cease all triggered autonomous reporting if it disassociates, or reassociates to a different BSS (reassociation to the same BSS shall not affect triggered reporting).
- FTM session, because it is analogous to triggered autonomous reporting.

- DMG SP allocation and CBAP allocation (pseudo-static or dynamic) – treat the same as TSes. (Note, the dynamic allocations lifetime is limited to a beacon interval, which barely makes sense across a Reassociation)

**Proposed changes** (referenced to D2.8):

At P1598.51, at the end of bullet (c), insert the following:

If the MLME-REASSOCIATION.request has the new AP’s BSSID in the CurrentAPAddress parameter (reassociation to the same AP), the following states, agreements and allocations shall be deleted or reset to initial values: all EDCAF state, any Block Ack agreements, sequence number, packet number, duplicate detection caches, anything queued for transmission, fragmentation and reassembly buffers, power management mode, and WNM Sleep mode. The following states, agreements and allocations are not affected by the reassociation procedure: PSMP sessions, Enablement/Deenablement, GDD enablement, STSL, DLS and TDLS agreements, SMKSAs, STKSAs and TPKSAs established with any peers, MMSLs, GCR agreements, DMS agreements, TFS agreements, FMS agreements, Triggered autonomous reporting agreements, FTM sessions, and DMG SP and CBAP allocations.

At P1618.52, change “A non-AP and non-PCP STA that associates, disassociates or re-associates shall locally delete all existing allocations and all TSs that have been established using a PTP TSPEC.” by adding, “(except for reassociation to the same AP)” following “re-associates”. Also change “re-associates” to “reassociates”.

At P1618.56, change “An AP or PCP that receives an MLME-ASSOCIATE.indication, MLME-REASSOCIATE.indication or MLME-DISASSOCIATE.indication primitive” by adding, “(except when the MLME-REASSOCIATE.indication has this AP’s BSSID in the CurrentAPAddress parameter)” following “MLME-REASSOCIATE.indication.

At P1619.5, change “A STA that generates an MLME-ASSOCIATE.request, MLME-REASSOCIATE.request or MLME-DISASSOCIATE.request primitive” by adding “(except when the MLME-REASSOCIATE.request has the target AP’s BSSID in the CurrentAPAddress parameter)” following “MLME-REASOCCIATE.request.”