IEEE P802.11
Wireless LANs

|  |
| --- |
| CID 8085 8031 resolution text |
| Date: 2016-07-26 |
| Author(s): |
| Name | Company | Address | Phone | email |
| Kazuyuki Sakoda | Sony |  |  | Kazuyuki.Sakoda (at) am (dot) sony (dot) com |
|  |  |  |  |  |
|  |  |  |  |  |

Abstract

This document provides suggested resolution to CID 8085 and 8031.

***CID 8085:***

***Comment****:*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 8085 | 3163.50 | C.3 |  | dot11MCCAMinTrackStates is "This is a capability variable.It is written by an external management entity." --- which is it? | Delete "It is written by an external management entity." in the cited text and also at 3164.6 | GEN |

***Discussion****:*

dot11MCCAMinTrackStates and dot11MCCAMinTrackStates are used to define trackable MCCAOP reservations for MCCA operation in MBSS.

The intended specification is as follows:

1. dot11MCCAMaxTrackStates is a capability variable. This value specifies the absolute maximum number of MCCAOP reservations that the device is able to track. This is a read-only variable, and the device cannot track MCCAOP reservations beyond this value in any case.
2. dot11MCCAMinTrackStates is a control variable that an administrator could manage the device to limit the number of MCCAOP reservations to track, i.e., to allow shared RAM resource management.
dot11MCCAMinTrackStates represents effective MCCAOP reservation the STA tracks.
3. dot11MCCAMaxTrackStates can be any number between 83 and 65535. 83 is the minimal number that the MCCA capable 802.11 device needs to track MCCAOP reservations.
4. dot11MCCAMinTrackStates can be set to any number between 83 and dot11MCCAMaxTrackStates.
5. STAs only track up to dot11MCCAMinTrackStates MCCAOP reservations. So, if the number of MCCAOP reservation that the STA tracks exceeds dot11MCCAMinTrackStates, the STA does not set up additional MCCAOP reservations. Other decision based on the number of tracking MCCAOP reservation will be made by referring to dot11MCCAMinTrackStates.

As the variables dot11MCCAMinTrackStates and dot11MCCAMaxTrackStates are poorly named, it is suggested to replace with dot11MCCATrackStatesActive and dot11MCCATrackStatesCapable.

***Proposed changes****:*

* Semantics of the service primitive

The primitive parameters are as follows:

MLME-ACTIVATEMCCA.request(

MCCAScanDuration,

MAFLimit,

MCCAAdvertPeriodMax,

MCCAMaxTrackStates,

MCCACWmin,

MCCACWmax,

MCCAAIFSN

)

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid range | Description |
| MCCAScanDuration | Integer | 0 – 65 535 | Specifies the duration in TUs that the mesh STA shall not initiate or accept MCCA Setup Request frames. |
| MAFLimit | Integer | 0–255 | Specifies the maximum MCCA access fraction allowed at the mesh STA. This number is always a multiple of (1/255) of the DTIM Interval. |
| MCCAAdvertPeriodMax | Integer | 0–255 | Specifies the maximum interval that a mesh STA with dot11MCCAActivated equal to true waits for an MCCAOP advertisement. It is expressed in number of DTIM intervals. |
| MCCAMaxTrackStates | Integer | 83– dot11MCCATrackStatesCapable  | Specifies the total number of MCCAOP reservations that the MAC entity is able to track. |
| MCCACWmin | Integer | 0–15 | Specifies the value of the minimum size of the contention window that the MAC entity uses for channel access during an MCCAOP. |
| MCCACWmax | Integer | 0–63 | Specifies the value of the maximum size of the contention that the MAC entity uses for channel access during an MCCAOP. |
| MCCAAIFSN | Integer | 0–15 | Specifies the value of the AIFSN that the MAC entity uses for channel access during an MCCAOP. |

* MCCAOP Setup Reply element

...

|  |
| --- |
| * MCCA Reply Code field values
 |
| MCCA reply code | Meaning |
| 0 | Accept |
| 1 | Reject: MCCAOP reservation conflict |
| 2 | Reject: MAF limit exceeded |
| 3 | Reject: MCCA track limit (dot11MCCATrackStatesActive) exceeded |
| 4–255 | Reserved |

…

* MCCAOP reservations

…

A mesh STA with dot11MCCAActivated equal to true shall be able to track at least dot11MCCATrackStatesActive MCCAOP reservations, including its own reservations. If the number of tracked MCCAOP reservations is less than dot11MCCATrackStatesActive, the mesh STA shall be able to track, set up, and accept additional reservations. In this case, the mesh STA shall set the Accept Reservations subfield in the Flags field to 1 in the MCCAOP Advertisement Overview elements it transmits.

If the number of tracked MCCAOP reservations is greater than or equal to(#3447) dot11MCCATrackStatesActive, the mesh STA shall not track, set up, or accept additional reservations. In this case, the mesh STA shall set the Accept Reservations subfield in the Flags field to 0 in the MCCAOP Advertisement Overview elements it transmits. Moreover, it shall reply to MCCA Setup Request frames with an MCCA Setup Reply frame with the MCCA Reply Code field in the MCCAOP Setup Reply element equal to 3: Reject: MCCAOP track limit exceeded.

The tracked MCCAOP reservations are advertised as described in 10.23.3.7 (MCCAOP advertisement). How to access the medium during the tracked MCCAOP reservations is specified in 10.23.3.9 (Access during MCCAOPs).

* MCCAOP setup procedure

…

* The MCCAOP responder shall verify the following conditions:
* The reservation does not overlap with its neighborhood MCCAOP periods.
* The reservation does not cause the MAF limit to be exceeded for itself or its neighbor mesh STAs.
* The number of reservations in its neighborhood MCCAOP periods does not exceed dot11MCCATrackStatesActive.

…

* Construction of an MCCAOP advertisement set

…

* The mesh STA constructs one MCCAOP Advertisement Overview element such that
* The MCCAOP Advertisement Set Sequence Number field is set to the advertisement set sequence number as explained in 10.23.3.7.3 (Setting the MCCAOP advertisement set sequence number).
* The Medium Access Fraction field is set to the medium access fraction.
* The MAF limit field is set to (#7128)dot11MAFlimit.
* The Accept Reservations field is set to 1 if the number of tracked reservations of this mesh STA is less than dot11MCCATrackStatesActive, and set to 0 otherwise.
* Bit *i* of the Advertisement Elements Bitmap field is set to 1 if an MCCAOP Advertisement element with the MCCAOP Advertisement Element Index subfield equal to *i* is part of the representation of this MCCAOP advertisement set, and set to 0 otherwise.
* MIB Detail

Dot11MeshSTAConfigEntry ::=

 SEQUENCE {

 dot11MeshID OCTET STRING,

 dot11MeshNumberOfPeerings Unsigned32,

 dot11MeshAcceptingAdditionalPeerings TruthValue,

 dot11MeshConnectedToMeshGate TruthValue,

 dot11MeshSecurityActivated TruthValue,

 dot11MeshActiveAuthenticationProtocol INTEGER,

 dot11MeshMaxRetries Unsigned32,

 dot11MeshRetryTimeout Unsigned32,

 dot11MeshConfirmTimeout Unsigned32,

 dot11MeshHoldingTimeout Unsigned32,

 dot11MeshConfigGroupUpdateCount Unsigned32,

 dot11MeshActivePathSelectionProtocol INTEGER,

 dot11MeshActivePathSelectionMetric INTEGER,

 dot11MeshForwarding TruthValue,

 dot11MeshTTL Unsigned32,

 dot11MeshGateAnnouncements TruthValue,

 dot11MeshGateAnnouncementInterval Unsigned32,

 dot11MeshActiveCongestionControlMode INTEGER,

 dot11MeshActiveSynchronizationMethod INTEGER,

 dot11MeshNbrOffsetMaxNeighbor Unsigned32,

 dot11MBCAActivated TruthValue,

 dot11MeshBeaconTimingReportInterval Unsigned32,

 dot11MeshBeaconTimingReportMaxNum Unsigned32,

 dot11MeshDelayedBeaconTxInterval Unsigned32,

 dot11MeshDelayedBeaconTxMaxDelay Unsigned32,

 dot11MeshDelayedBeaconTxMinDelay Unsigned32,

 dot11MeshAverageBeaconFrameDuration Unsigned32,

 dot11MeshSTAMissingAckRetryLimit Unsigned32,

 dot11MeshAwakeWindowDuration Unsigned32,

 dot11MCCAImplemented TruthValue,

 dot11MCCAActivated TruthValue,

 dot11MAFlimit Unsigned32,

 dot11MCCAScanDuration Unsigned32,

 dot11MCCAAdvertPeriodMax Unsigned32,

 dot11MCCATrackStatesCapable Unsigned32,

 dot11MCCATrackStatesActive Unsigned32,

 dot11MCCAOPtimeout Unsigned32,

 dot11MCCACWmin Unsigned32,

 dot11MCCACWmax Unsigned32,

 dot11MCCAAIFSN Unsigned32

 }

dot11MCCATrackStatesCapable OBJECT-TYPE

 SYNTAX Unsigned32 (83..65535)

 MAX-ACCESS read-only

 STATUS current

 DESCRIPTION

 "This is a capability variable.

Its value is determined by device capabilities.

 This attribute specifies the number of MCCAOP reservations that the MAC entity is able to track."

DEFVAL { 83 }

::= { dot11MeshSTAConfigEntry 35 }

dot11MCCATrackStatesActive OBJECT-TYPE

 SYNTAX Unsigned32 (83.. dot11MCCATrackStatesCapable)

 MAX-ACCESS read-write

 STATUS current

 DESCRIPTION

 "This is a control variable.

 It is written by an external management entity.

 Changes take effect as soon as practical in the implementation.

 The upper bound is given by the current value of dot11MCCATrackStatesCapable.

 This attribute specifies the maximum number of MCCAOP reservations that the MAC entity is able to track."

 DEFVAL { 83 }

 ::= { dot11MeshSTAConfigEntry 36 }

dot11MeshOptionGroup OBJECT-GROUP

 OBJECTS {

 -- dot11MeshSTAConfigTable

 dot11MeshConfigGroupUpdateCount,

 dot11MeshGateAnnouncementInterval,

 dot11MeshBeaconTimingReportInterval,

 dot11MeshBeaconTimingReportMaxNum,

 dot11MeshDelayedBeaconTxInterval,

 dot11MeshDelayedBeaconTxMaxDelay,

 dot11MeshDelayedBeaconTxMinDelay,

 dot11MeshAverageBeaconFrameDuration,

 dot11MeshSTAMissingAckRetryLimit,

 dot11MeshAwakeWindowDuration,

 dot11MAFlimit,

 dot11MCCAScanDuration,

 dot11MCCAAdvertPeriodMax,

 dot11MCCATrackStatesCapable,

 dot11MCCATrackStatesActive,

 dot11MCCAOPtimeout,

 dot11MCCACWmin,

 dot11MCCACWmax,

 dot11MCCAAIFSN

 }

 STATUS current

 DESCRIPTION

 "This object class provides the objects from the IEEE Std 802.11 MIB required to manage optional mesh functionality. Note that other objects for managing mesh functionality are located in the dot11MeshComplianceGroup, dot11MeshHWMPComplianceGroup, and dot11PasswordAuthComplianceGroup."

 ::= { dot11Groups 60 }

***CID 8031****:*

***Comment****:*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Page** | **Clause** | **Comment** | **Proposed Change** |
| 8031 | 2148.01 | 14.7 | Subclause 14.7 (Mesh Security) describes summary of the mesh security operation. The 2nd paragaraph of 14.7 is very confusing and does not reflect normative behavior. It is better to refine the 2nd paragraph. | Replace"When dot11MeshSecurityActivated is true, all mesh Data frames and individually addressed Management frames (excluding Authentication frames and self-protected Management frames) shall be protected by the mesh TKSA, and all group addressed Data frames and group addressed Management frames that are indicated as "Group Addressed Privacy" in Table 9-47 (Category values) shall be protected by the mesh GTKSA."with"When dot11MeshSecurityActivated is true, all individually addressed mesh Data frames and robust Management frames (see 12.2.8 (Requirements for robust management frame protection)) shall be protected by the mesh TKSA, and all group addressed mesh Data frames and Action frames that are indicated as "Group Addressed Privacy" in Table 9-47 (Category values) shall be protected by the mesh GTKSA.When dot11RSNAProtectedManagementFramesActivated is true, group addressed robust Management frames that are not protected by the mesh GTKSA shall be protected using BIP (see 11.13 (Group addressed robust management frame procedures))." |

***Discussion****:*

Subclause 14.7 mesh security is supposed to present enough information on mesh security, but the subclause does not describe whole picture of the mesh security operation.

The intended normative behaviours are as follows:

1. When dot11MeshSecurityActivated is true, individually addressed mesh Data frames and robust Management frames are encrypted using mesh TKSA (see 14.7 Mesh security).





1. When dot11MeshSecurityActivated is true, group addressed mesh Data frames and Action frames indicated as “Group Addressed Privacy” in Table 9-47 are encrypted using mesh GTKSA. (see 11.13 Group addressed robust management frame procedures)



1. When dot11RSNAProtectedManagementFramesActivated is true, group addressed robust Management frames that are not encrypted using mesh GTKSA shall be protected using BIP using IGTK negotiated through Authenticated Mesh Peering Exchange (see 9.4.2.118 Authenticated Mesh Peering Exchange element)





***Proposed changes****:*

**14.7 Mesh security**

During the AMPE, the peers negotiate, and agree upon, a pairwise ciphersuite and a group cipher suite. They also establish a mesh TKSA and mesh GTKSA to be used with the pairwise cipher suite and group cipher suite, respectively.

When dot11MeshSecurityActivated is true, all individually addressed mesh Data frames and individually addressed robust Management frames (see 12.2.8 (Requirements for robust management frame protection)) shall be protected by the mesh TKSA, and all group addressed Data frames and group addressed Action frames that are indicated as “Group Addressed Privacy” in Table 9-47 (Category values) shall be protected by the mesh GTKSA.

When dot11RSNAProtectedManagementFramesActivated is true, group addressed robust Management frames that are not protected by the mesh GTKSA shall be protected using BIP (see 11.13 (Group addressed robust management frame procedures)).

**Reference**:

[1] Draft P802.11REVmc\_D6.0.