**IEEE P802.11
Wireless LANs**

|  |
| --- |
| **802.11****CR 2693 Mirrored SCS** |
| **Date:** 2019-03-11 |
| **Author(s):** |
| **Name** | **Affiliation** | **Address** | **Phone** | **Email** |
| Thomas Derham | Broadcom | 16340 W Bernardo Dr, San Diego CA |  | thomas.derham@broadcom.com |
| Matthew Fischer | Broadcom | 270 Innovation Drive, San Jose CA |  |  |
| Sriram Neelakandan | Broadcom | S1, Wipro E-City SEZ, Bangalore |  |  |
| Neeraj Garg | Broadcom |  |  |  |

**Abstract**

This document provides comment resolutions for REVmd letter ballot CID 2693, by defining a Mirrored SCS capability. The document is based on REVmd D2.1.

R0: Initial draft

R1: Addressed comments – introduced stream timeout so that AP does not need to continue to maintain state for tracking old streams, introduced ability for AP to suggest parameters when tearing down / rejecting an MSCS request, revised text to clarify how UPs of streams are determined, updated and additional examples in the discussion, various clean-up. Redline is with respect to R0

R2: Addressed offline comment regarding MSDU reordering

R3: Addressed offline comment to add MLME primitives and clarify MSDU classification occurs above the MAC (similar to TS operation)

**Addressed Comments**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CID** | **Comment** | **Page number**  | **Subclause** | **Line number (wrt D4.0)** | **Proposed Change by commenter** | **Resolution** |
| 2693 | When non-AP STA sends SCS Request to AP in order to request the AP to assign a specified UP to certain MSDUs it transmits (e.g. downstream packet stream to that STA), the STA needs to specify one or more TCLAS classifiers (e.g. IP 5-tuples), which can be complex for STA to determine - e.g. a mobile app may interact with multiple internet/web servers with dynamic IPs/ports | 2418 | 11.26.2 | 58 | Allow SCS to support a "reflective" mode where the TCLAS defines just the Classifier Type and Classifier Mask. Example: the AP categorizes upstream SCS streams based on that mask, and derives the UP for each corresponding downstream SCS stream from the UP of the upstream | Resolve in the direction of the commenter’s proposal, per amendments in this document. |

**Discussion**

Stream Classification Service (SCS) enables a non-AP STA to request to its associated AP that specific QoS treatment (setting of UP, alternate EDCA queue and/or drop eligibility) is applied to unicast MSDUs classified as a particular stream, where that stream comprises MSDUs that are incoming to the AP (and after processing, will be outgoing) that match parameters specified in one or more TCLAS elements.

SCS can be used in both LAN-based use cases (e.g. streaming from a local media server to a video endpoint) and WAN-based (e.g. internet, enterprise WAN) use cases. For example, an audio-visual or gaming mobile app on a non-AP STA might identify a QoS-sensitive AV or metadata internet stream based on IP layer classification parameters (e.g. source/destination IP address, port), and request the AP to apply specific QoS treatment to that stream from the AP to the STA. (The non-AP STA can apply QoS treatment for the uplink direction of its own accord). Note that, for downlink Internet traffic, it is often the case that layer-3 QoS markings (e.g. DSCP/TOS) that may be applied by the source server are stripped/overwritten by intermediate routers, and therefore local QoS assignment at ingress to the local network is often necessary.

However, in some mainstream use cases, it is challenging for a non-AP STA to identify the parameters needed to classify a stream with SCS. In the example above, the application layer might only be aware of the FQDN of the internet server (which is the source of the stream for which QoS treatment is required); its IP address might dynamically change due to (for example) DNS load balancing and the STA might not have suitable framework APIs to obtain/correlate this information across the stack. Further, the application might dynamically establish multiple sessions with multiple (server) endpoints in real-time, e.g. as certain features are used within the application (e.g. chat, presence, AR/VR features, multi-stream video, etc), and requesting SCS treatment for new streams - using parameters that may not be available until the stream is about to commence - can be burdensome and insufficiently responsive.

The Traffic Stream (TS) operation features (11.4) provide additional and/or complementary capabilities compared to SCS (e.g. resource reservation), however they have the same issues as SCS in terms of stream classification definition.

Therefore, this document introduces a variant of SCS called Mirrored SCS (MSCS) to address these use cases. MSCS, as is the case for SCS, is initiated by a non-AP STA that requests the AP to apply QoS treatment (setting of UP) to unicast MSDUs destined to that non-AP STA based on classification parameters. The main differences between MSCS and SCS are as follows:

* In SCS, a single SCS Descriptor corresponds to a request to the AP to classify a single stream (identified by an SCSID) based on specific classification values (e.g. source IP=50.1.1.1 port=443); whereas in MSCS, the MSCS Descriptor specifies the set of classification parameters (e.g. source IP, port), not the actual values of those parameters. Therefore, in general, MSCS results in the AP identifying and tracking multiple streams (which do not have explicit identifiers), where the MSDUs comprising each stream have identical values for the specified parameters
* In SCS, the QoS to be assigned to classified MSDUs of a given stream is explicitly specified by the non-AP STA (e.g. UP=6, AltEDCAQueue=0, DE=0); whereas in MSCS, the QoS (UP) to be assigned to MSDUs for each of the streams identified by the AP is implicitly derived from the UP of MSDUs sent in the corresponding reverse stream from the non-AP STA to the AP
* In SCS the streams are not necessarily bound to the link between the requesting non-AP STA and the AP; whereas in MSCS the derivation of UP from the reverse stream dictates that classification is bound to streams between the AP and the requesting non-AP STA

The following is an example of a typical expected MSCS use case:

* 1. An end-user opens an application on a non-AP STA that interacts with web servers on the public Internet and has traffic flows that require certain QoS treatment.
* 2. The non-AP STA sends an MSCS Request with Request Type set to “Add”, where the User Priority control field indicates UPs {4, 5, 6, 7} and UP limit of 7, Stream Timeout corresponding to 60 seconds, and a TCLAS Mask element containing the specified list of classification parameters (type=4, Classifier Mask = {Source IP address, Source Port}).
* 3. The application begins initiating various HTTP sessions with the web servers. The AP monitors the upstream MSDUs received from the non-AP STA and, for those MSDUs that have UP of 4, 5, 6 or 7 (i.e. upstream packets with AC=VI or AC=VO), adds or updates an entry in its *UP{tuple}* list where *tuple* is the tuple of classification parameters for the corresponding downstream MSDUs in the same stream. For example:
	+ when the non-AP STA sends an MSDU with UP=6 with destination IP address = 123.1.1.1 and destination port = 80, the AP adds a variable to the list mapped to tuple {source IP address = 123.1.1.1, source port = 80} with value of 6
	+ when the non-AP STA sends an MSDU with UP=4 with destination IP address = 123.1.1.2 and destination port = 443, the AP adds a variable to the list mapped to tuple {source IP address = 123.1.1.2, source port =443} with value of 4
* 4. In parallel, the AP monitors incoming MSDUs destined to the non-AP STA. When an MSDU matches a value of *tuple* for a variable in the list, it sets the UP of that MSDU to the value of that variable in the list, thus “mirroring” the QoS of an uplink stream to the corresponding downlink stream. For the example above:
	+ when an incoming MSDU destined to the non-AP STA has {source IP address = 123.1.1.1, source port = 80}, the AP sets the UP of that MSDU to the value of 6
	+ when an incoming MSDU destined to the non-AP STA has {source IP address = 123.1.1.2, source port = 443}, the AP sets the UP of that MSDU to the value of 4
* 5. The application finishes interacting with the web server and so no further packets flow that match the tuples in the list. The AP deletes those entries from its list once the timeout value is reached.

The following are some examples of MSCS negotiation and adaption between the non-AP STA and AP:

* Ex a). The AP does not have an active MSCS for a non-AP STA, and receives an MSCS request from that non-AP STA for which it does not support the specified TCLAS Mask type or specified combination of classifier parameters or timeout parameter. The AP rejects the request with status code “REQUESTED\_TCLAS\_NOT\_SUPPORTED”, and might include a suggested set of (similar) parameters that the AP would be able to support in a subsequent request.
* Ex b). The AP does not have an active MSCS for a non-AP STA, but does have other strict QoS policies configured for that non-AP STA (e.g. DSCP or 802.1Q mapping policies) and receives an MSCS request from that non-AP that might cause conflict with those policies. The AP rejects the request with status code “TCLAS\_PROCESSING\_TERMINATED\_POLICY\_CONFLICT”
* Ex c). The AP is operating active MSCSs for certain associated non-AP STAs, and receives an MSCS request from another non-AP STA for which it does not have sufficient processing resources to support. The AP rejects the request with status code “INSUFFICIENT\_TCLAS\_PROCESSING\_RESOURCES”
* Ex d). The AP accepts an MSCS request from a non-AP STA, which then initiates multiple streams with web servers using high UP values in the uplink. These streams comprise a large volume of downlink packets which (per the MSCS parameters) the AP is allocating high UP values. The AP determines that this is causing unacceptable impact on network management (e.g. airtime fairness, spectral efficiency) and decides to tear-down the active MSCS with status code “TCLAS\_PROCESSING\_TERMINATED\_INSUFFICIENT\_QOS”, and might include a suggested set of (similar) parameters that the AP would be able to support in a subsequent request (e.g. with lower User Priority Limit value).

***Instruct the editor to add the following section:***

**4.3.24.2a Mirrored stream classification service (MSCS)**

MSCS enables the establishment of classification using layer 2 and/or layer 3 signaling to classify incoming unicast MSDUs into streams. Once classified, unicast MSDUs in each stream are assigned to a user priority based on the user priority of MSDUs matching the stream in the reverse direction.

* **Extended Capabilities element**

***Instruct the editor to add the following row to Extended Capabilities table as follows:***

|  |  |  |
| --- | --- | --- |
| (#1284)82 | SAE Password Identifiers Used Exclusively | The AP sets the SAE Password Identifiers Used Exclusively field(Ed) to 1 when every password in the dot11RSNAConfigPasswordValueTable(Ed) has a password identifier and sets it to 0 otherwise. See 12.4.3 (Representation of a password). |
| ANA | Mirrored SCS | The STA sets the Mirrored SCS field to 1 when dot11MSCSActivated is true and sets it to 0 otherwise. |
| (11ai)(Ed)~~83~~ANA–*n* | Reserved |  |

***Instruct the editor to add the following section at the end of 6.3:***

6.3.xx MSCS request and response procedure

6.3.xx.1 General

The following MLME primitives support the signaling of the MSCS request and response procedure.

6.3.xx.2 MLME-MSCS.request

6.3.xx.2.1 Function

This primitive requests transmission of an MSCS Request frame to an AP.

6.3.xx.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-MSCS.request(

PeerSTAAddress,
DialogToken,
MSCSDescriptor,
VendorSpecificInfo
)

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid range | Description |
| PeerSTAAddress | MAC Address | Any valid individual MAC address | Specifies the address of the peer MAC entity with which to perform the MSCS process. |
| DialogToken | Integer | 1–255 | The dialog token to identify the MSCS request and response transaction. |
| MSCSDescriptor | MSCS Descriptor element | MSCS Descriptor element, as defined in 9.4.2.xxx (MSCS Descriptor element) | Specifies classifiers and parameters for the MSCS. |
| VendorSpecificInfo | A set of elements | As defined in 9.4.2.25 (Vendor Specific element) | Zero or more elements. |

6.3.xx.2.3 When generated

This primitive is generated by the SME to request that a MSCS Request frame be sent to the AP with which the STA is associated.

6.3.xx.2.4 Effect of receipt

On receipt of this primitive, the MLME constructs a MSCS Request frame. The STA then attempts to transmit this frame to the AP with which the STA is associated.

6.3.xx.3 MLME-MSCS.confirm

6.3.xx.3.1. Function

This primitive reports the result of a MSCS procedure.

6.3.xx.3.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-MSCS.confirm(

PeerSTAAddress,
DialogToken,
Status,

MSCSDescriptor
VendorSpecificInfo
)

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid range | Description |
| PeerSTAAddress | MAC Address | Any valid individual MAC address | Specifies the address of the peer MAC entity with which to perform the MSCS process. |
| DialogToken | Integer | 1–255 | The dialog token to identify the MSCS request and response transaction. |
| Status | Enumeration(Ed) | See Table 9-52 (Status codes) | Indicates the result response. See Table 9-52 (Status codes). |
| MSCSDescriptor | MSCS Descriptor element | MSCS Descriptor element, as defined in 9.4.2.xxx (MSCS Descriptor element) | Zero or one elements.Specifies suggested classifiers and parameters for the MSCS. |
| VendorSpecificInfo | A set of elements | As defined in 9.4.2.25 (Vendor Specific element) | Zero or more elements. |

6.3.xx.3.3 When generated

This primitive is generated by the MLME as a result of an MLME-MSCS.request primitive and indicates the results of the request.

This primitive is generated when the STA receives a MSCS Response frame from the AP.

6.3.xx.3.4 Effect of receipt

On receipt of this primitive, the SME should operate according to the procedure in 11.26.3 (MSCS procedures).

6.3.xx.4 MLME-MSCS.indication

6.3.xx.4.1 Function

This primitive indicates that an MSCS Request frame was received from a non-AP STA.

6.3.xx.4.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-MSCS.indication(

PeerSTAAddress,
DialogToken,
MSCSDescriptor,
VendorSpecificInfo
)

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid range | Description |
| PeerSTAAddress | MACAddress | Any valid individual MAC address | The address of the non-AP STA MAC entity from which an MSCS Request frame was received. |
| DialogToken | Integer | 1–255 | The dialog token to identify the MSCS request and response transaction. |
| MSCSDescriptor | MSCS Descriptor element | MSCS Descriptor element, as defined in 9.4.2.xxx (MSCS Descriptor element) | Specifies classifiers and parameters for the MSCS. |
| VendorSpecificInfo | A set of elements | As defined in 9.4.2.25 (Vendor Specific element) | Zero or more elements. |

6.3.xx.4.3 When generated

This primitive is generated by the MLME when an MSCS Request frame is received.

6.3.xx.4.4 Effect of receipt

On receipt of this primitive, the SME should operate according to the procedure in 11.26.3 (MSCS procedures).

6.3.xx.5 MLME-MSCS.response

6.3.xx.5.1 Function

This primitive is generated in response to an MLME-MSCS.indication primitive requesting an MSCS Response frame be sent to a non-AP STA.

6.3.xx.5.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-MSCS.response(

PeerSTAAddress,
DialogToken,

Status,

MSCSDescriptor,
VendorSpecificInfo
)

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid range | Description |
| PeerSTAAddress | MACAddress | Any valid individual MAC address | The address of the non-AP STA MAC entity from which a MSCS Request frame was received. |
| DialogToken | Integer | 1–255 | The dialog token to identify the MSCS request and response transaction. |
| Status(#1567) | Enumeration(Ed) | See Table 9-52 (Status codes) | Indicates the result response. See Table 9-52 (Status codes). |
| MSCSDescriptor | MSCS Descriptor element | MSCS Descriptor element, as defined in 9.4.2.xxx (MSCS Descriptor element) | Zero or one elements.Specifies suggested classifiers and parameters for the MSCS. |
| VendorSpecificInfo | A set of elements | As defined in 9.4.2.25 (Vendor Specific element) | Zero or more elements. |

6.3.xx.5.3 When generated

This primitive is generated by the SME in response to an MLME-MSCS.indication primitive requesting an MSCS Response frame be sent to a non-AP STA.

6.3.xx.5.4 Effect of receipt

On receipt of this primitive, the MLME constructs a MSCS Response frame. The STA then attempts to transmit this frame to the non-AP STA indicated by the PeerSTAAddress parameter.

6.3.xx.6 MLME-SCS-TERM.request

6.3.xx.6.1 Function

This primitive requests the transmission of an MSCS Response frame to a non-AP STA to terminate an active MSCS.

6.3.xx.6.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-MSCS-TERM.request(

PeerSTAAddress,
DialogToken,
Status,

MSCSDescriptor,
VendorSpecificInfo
)

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid range | Description |
| PeerSTAAddress | MACAddress | Any valid individual MAC address | The address of the non-AP STA MAC entity to which the MSCS Response frame is to be sent. |
| DialogToken | Integer | 0 | Set to 0 for an autonomous MSCS Response frame. |
| Status | Enumeration(Ed) | See Table 9-52 (Status codes) | Indicates the result response. See Table 9-52 (Status codes). |
| MSCSDescriptor | MSCS Descriptor element | MSCS Descriptor element, as defined in 9.4.2.xxx (MSCS Descriptor element) | Zero or one elements.Specifies suggested classifiers and parameters for the MSCS. |
| VendorSpecificInfo | A set of elements | As defined in 9.4.2.25 (Vendor Specific element) | Zero or more elements. |

6.3.xx.6.3 When generated

This primitive is generated by the SME to terminate an active MSCS.

6.3.xx.6.4 Effect of receipt

On receipt of this primitive, the MLME constructs an MSCS Response frame. The STA then attempts to transmit this frame to the non-AP STA indicated by the PeerSTAAddress parameter.

6.3.xx.7 MLME-MSCS-TERM.indication

6.3.xx.7.1 Function

This primitive is generated by the MLME when an unsolicited MSCS Response frame is received.

6.3.xx.7.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-MSCS-TERM.indication(

ResultCode,
DialogToken,
Status,

MSCSDescriptor,
VendorSpecificInfo
)

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Valid range | Description |
| ResultCode | Enumeration | SUCCESS, FAILURE | Indicates the result of the MLME-MSCS-TERM.request primitive. |
| DialogToken | Integer | 0 | Set to 0 for an autonomous MSCS Response frame. |
| Status | Enumeration(Ed) | See Table 9-52 (Status codes) | Indicates the result response of the MSCS. See Table 9-52 (Status codes). |
| MSCSDescriptor | MSCS Descriptor element | MSCS Descriptor element, as defined in 9.4.2.xxx (MSCS Descriptor element) | Zero or one elements.Specifies suggested classifiers and parameters for the MSCS. |
| VendorSpecificInfo | A set of elements | As defined in 9.4.2.25 (Vendor Specific element) | Zero or more elements. |

6.3.xx.7.3 When generated

This primitive is generated when the STA receives an unsolicited MSCS Response frame from the AP.

6.3.xx.7.4 Effect of receipt

On receipt of this primitive, the SME should operate according to the procedure in 11.26.3 (MSCS procedures).

***Instruct the editor to add the following section, and add the element to Element ID table:***

**9.4.2.xxx** **TCLAS Mask element**

The TCLAS Mask element contains a set of parameters necessary to classify incoming MSDUs into streams based on a classifier mask. The structure of this element is shown in Figure 9-302 (TCLAS Mask element format).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Element ID | Length | Element ID Extension | Frame Classifier |
| Octets: | 1 | 1 | 1 | variable |
| **Table 9-xx** | **TCLAS Mask element format** |

The Element ID, Length and Element ID Extension fields are defined in 9.4.2.1 (General).

The Frame Classifier field specifies the parameters that are used to classify incoming MSDUs into streams. The field is defined in 9.4.2.30 (TCLAS element) (see Figure 9-303 (Frame Classifier field)), except that, in the Classifier Parameters subfield, all subfields other than the following (if present) are reserved:

* Filter Offset subfield
* Filter Mask subfield(s) (including MAC Header Filters in Match Specification subfields)
* Previous Protocol Number of Next Header subfield

***Instruct the editor to add the following section, and add the element to Element ID table:***

**9.4.2.xxx MSCS Descriptor element**

The MSCS Descriptor element defines information about the parameters used to classify streams using the procedures defined in 11.26.3 (MSCS procedures). The format of the MSCS Descriptor element is shown in Figure 9-xxx (MSCS Descriptor element format).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Element ID | Length | Element ID Extension | Request Type | User Priority Control | Stream Timeout | TCLAS Mask Elements (optional) | Optional Subelements  |
| Octets: | 1 | 1 | 1 | 1 | 2 | 4 | variable | variable |

**Figure 9-XXX MSCS Descriptor element format**

The Element ID, Length and Element ID Extension fields are defined in 9.4.2.1 (General).

The Request Type field is as defined in 9.4.2.121 (SCS Descriptor element).

The User Priority Control field is shown in Figure 9-XXX User Priority Control field. This field is reserved when the Request Type field is “Remove”.

|  |  |  |  |
| --- | --- | --- | --- |
|  | B0 B7  | B8 B10 | B11 B15 |
|  | User Priority Bitmap | User Priority Limit | Reserved |
| Bits: | 8 | 3 | 5 |

The User Priority Bitmap subfield is one octet in length. Each bit in the bitmap corresponds to a user priority (UP), with the least significant bit corresponding to UP value of 0, and the most significant bit corresponding to UP value of 7. A value of 1 in a bit position in the bitmap indicates that the corresponding UP is used when assigning a UP to streams classified by MSCS; a value of 0 in a bit position indicates that the corresponding UP is not used for this purpose.

The User Priority Limit subfield is 3 bits in length and has a value between 0 and 7; it defines the maximum limit for the User Priority that is assigned to incoming MSDUs in the streams classified by MSCS.

The Stream Timeout subfield is 4 octets in length, and indicates the minimum timeout value, in TUs, for maintaining a variable *UP{tuple}* in the MSCS list. This subfield is reserved when the Request Type field is “Remove”.

The TCLAS Mask Elements field contains zero or more TCLAS Mask elements to specify how incoming MSDUs are classified into streams in MSCS, as defined in 9.4.2.xxx (TCLAS Mask element). One or more TCLAS Mask elements are present when the Request Type field is “Add” or “Change”; no TCLAS Mask elements are present when the Request Type field is “Remove.”

The Optional Subelements field is as defined in 9.4.2.121 (SCS Descriptor element).The MSCS Descriptor element is included in MSCS Request frames, as described in 9.6.18.6 (MSCS Request frame format), and in certain MSCS Response frames, as described in 9.6.18.7 (MSCS Response frame format). The use of the MSCS Descriptor element is described in 11.26.3 (MSCS procedures).

* **Robust AV Streaming Action frame details**
* **General**

***Instruct the editor to modify as follows***

Several Action frame formats are defined to support robust AV streaming. The Robust Action field values associated with each frame format within the robust AV streaming category are defined in Table 9-454 (Robust AV streaming Robust Action field values). The frame formats are defined in 9.6.18.2 (SCS Request frame format) to ~~9.6.18.5 (Group Membership Response frame format).~~9.6.18.7 (MSCS Response frame format).

|  |
| --- |
| * **Robust AV streaming Robust Action field values**
 |
| **Robust Action field value** | **Meaning** |
| 0 | SCS Request |
| 1 | SCS Response |
| 2 | Group Membership Request |
| 3 | Group Membership Response  |
| 4 | MSCS Request |
| 5 | MSCS Response |
| 6~~4~~–255 | Reserved |

***Instruct the editor to add the following sections, and also to add the following entries to Table 9-52 (Status Codes):***

***Name: “TCLAS\_PROCESSING\_TERMINATED\_INSUFFICIENT\_QOS”; Description: “Requested TCLAS processing has been terminated by the AP due to insufficient QoS capacity”***

***Name: “TCLAS\_PROCESSING\_TERMINATED\_POLICY\_CONFLICT”; Description: “Requested TCLAS processing has been terminated by the AP due to conflict with higher layer QoS policies”***

* + - 1. **MSCS Request frame format**

MSCS Request frames are used to request the creation, modification, or deletion of mirrored stream classification using the procedures defined in 11.26.3 (MSCS procedures).

The Action field of the MSCS Request frame contains the information shown in Figure 9-xxx (MSCS Request frame Action field format).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Category | Robust Action | Dialog Token | MSCS Descriptor element |
| Octets: | 1 | 1 | 1 | variable |
| **Figure 9-xxx MSCS Request frame Action field format** |

The Category field is defined in 9.4.1.11 (Action field).

The Robust Action field is defined in 9.6.18.1 (General).

The Dialog Token field is defined in 9.4.1.12 (Dialog Token field) and set by the requesting STA to a nonzero value that is used for matching action responses with action requests. See 10.29.5 (Operation of the Dialog Token field).

The MSCS Descriptor element is defined in 9.4.2.xxx (MSCS Descriptor element).

* + - 1. **MSCS Response frame format**

The MSCS Response frame is sent in response to an MSCS Request frame using the procedures defined in 11.26.3 (MSCS procedures). The Action field of an MSCS Response frame contains the information shown in Figure 9-xxx (MSCS Response frame Action field format).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Category | Robust Action | Dialog Token | Status | MSCS Descriptor element (optional) |
| Octets: | 1 | 1 | 1 | 2 | variable |
| **Figure 9-xxx MSCS Response frame Action field format** |  |

The Category field is defined in 9.4.1.11 (Action field).

The Robust Action field is defined in 9.6.18.1 (General).

The Dialog Token field is set to the nonzero value of the corresponding MSCS Request frame. If the MSCS Report frame is being transmitted for a reason other than in response to an MSCS Request frame, then the Dialog Token field is set to 0.

The Status field indicates the status of the request, as indicated in Table 9-52 (Status codes).

The MSCS Descriptor element is defined in 9.4.2.xxx (MSCS Descriptor element). When the Status field is “SUCCESS” the element is not present; otherwise it is optionally present as described in 11.26.3 (MSCS procedures).

* **Robust AV streaming**

***Instruct the Editor to add the following section:***

* + 1. **MSCS procedures**

The mirrored stream classification service (MSCS) is a service that may be provided by an AP to its associated STAs that support MSCS. In MSCS, the AP classifies incoming unicast MSDUs from the DS or WM that are destined for a non-AP STA based upon classifier masks provided by that non-AP STA. The AP sets the UP of the MSDUs in the classified streams based on the UP of unicast MSDUs in the corresponding mirror (reverse direction) stream from the non-AP STA to the AP.

Implementation of MSCS is optional for a STA. A STA that supports MSCS sets its dot11MSCSActivated to true, and shall set to 1 the Mirrored SCS field of the Extended Capabilities elements that it transmits. When dot11MSCSActivated is true, dot11QosOptionImplemented shall be true.

An MSCS setup is initiated by a non-AP STA’s SME. How it does this, and how it selects the MSCS Descriptor parameters, is out of scope of this standard. Setup of an MSCS uses the MLME primitives defined in 6.3.xx (MSCS request and response procedures).

A non-AP STA that supports MSCS may request use of MSCS, or request to update parameters of the currently active MSCS, by sending an MSCS Request frame that includes an MSCS Descriptor element with the Request Type field set to “Add” or “Change”, respectively. The MSCS Descriptor List field in the MSCS Descriptor element identifies how MSDUs are classified into streams and indicates parameters that determine the priority to assign to the classified MSDUs.

In a TCLAS Mask element in an MSCS Descriptor element, the Classifier Type subfield shall be set to a value that corresponds to a classifier of MSDUs, i.e. less than or equal to 5, or equal to 10.

Upon receipt of an MSCS Request frame from an associated non-AP STA, the AP shall respond with a corresponding MSCS Response frame. A value of “SUCCESS” shall be set in the Status field in the MSCS Response frame when the AP accepts the MSCS request. A value of “REQUEST\_DECLINED,” “REQUESTED\_TCLAS\_NOT\_SUPPORTED,” or “INSUFFICIENT\_TCLAS\_PROCESSING\_RESOURCES” shall be set in the Status field in the MSCS Response frame when the AP denies the MSCS request; an MSCS Descriptor element is optionally present in the response. If an MSCS Descriptor element is present, the Request Type field is set to “Change” and the element indicates a suggested set of parameters that could be accepted by the AP in response to a subsequent request by the non-AP STA. The AP shall decline an MSCS request with the Request Type field set to “Add” or “Change” if a TCLAS Mask element is not present.

An AP has a maximum of one active MSCS per non-AP STA. A non-AP STA shall not sent an MSCS request to an AP with the Request Type field set to “Add” if MSCS is currently active with that AP. An AP shall decline an MSCS request with the Request Type field set to “Add” if MSCS is currently active for the requesting non-AP STA.

If MSCS for a non-AP STA is currently active and the AP denies an MSCS request from the non-AP STA with Request Type field set to “Change”, the previously accepted parameters continue to apply.

If the request is accepted by the AP, MSCS for the non-AP STA becomes active. Incoming MSDUs whose DA parameter value maps to an RA equal to the MAC address of the non-AP STA are classified above the AP’s MAC sublayer and are sent to the MAC through the MAC SAP using the MA-UNITDATA.request primitive with the priority parameter equal to the assigned UP, based on the parameters specified in the MSCS Descriptor element. The following procedure is used:

a) The classifier parameters and associated masks of the MSCS are determined according to the TCLAS Mask element(s) in the MSCS Descriptor element as follows:

* For TCLAS Mask elements indicating a classifier type value less than or equal to 5, but not equal to 3, the classifier parameters specified by that element are indicated by the Classifier Mask subfield; there is no associated mask for these parameters
* For TCLAS Mask elements indicating classifier type 3, the classifier parameter specified by that element is equal to the set of octets defined by the Filter Offset subfield and length of the (reserved) Filter Value subfield; the associated mask for the parameter is specified in the Filter Mask subfield
* For TCLAS Mask elements indicating classifier type 10, the classifier specified by that element is equal to the set of octets defined by the Protocol Number or Next Header subfield and the length of the (reserved) Filter Value subfield; the associated mask for the parameter is specified in the Filter Mask subfield
* The classifier parameters and associated masks of the MSCS comprise all the classifier parameters and associated masks specified by all the TCLAS Mask elements in the MSCS Descriptor element.

b) The mirror classifier parameters and associated masks are determined as follows:

* If source (Ethernet) address is a classifier parameter, destination (Ethernet) address is a mirror classifier parameter; there is no associated mask for this parameter
* If destination (Ethernet) address is a classifier parameter, source (Ethernet) address is a mirror classifier parameter; there is no associated mask for this parameter
* If source IP address is a classifier parameter, destination IP address is a mirror classifier parameter; there is no associated mask for this parameter
* If destination IP address is a classifier parameter, source IP address is a mirror classifier parameter; there is no associated mask for this parameter
* If source port is a classifier parameter, destination port is a mirror classifier parameter; there is no associated mask for this parameter
* If destination port is a classifier parameter, source port is a mirror classifier parameter; there is no associated mask for this parameter
* If any other parameter is a classifier parameter, the same parameter is a mirror classifier parameter; if the classifier parameter has an associated mask then the mirror classifier parameter has the same associated mask

c) A logical list is maintained, initially empty, of variables *UP{tuple}*, where each variable represents a user priority associated with a value *tuple* of the masked classifier parameters. Monitoring of incoming unicast MSDUs as indicated by the MA-UNITDATA.indication primitive begins. If an MSDU is received that is not part of a TS (as described in 11.4 (TS operation)) and the SA parameter value is equal to the MAC address of the non-AP STA and the priority parameter is equal to an integer value in the range 0 to 7 and the service class parameter is equal to QoSAck, the MSDU’s UP is determined per 5.1.1.3 (Interpretation of priority parameter in MAC service primitives) and, if the UP is equal to one of the UPs specified in the User Priority bitmap subfield of the MSCS Descriptor element, the following steps are performed:

* (1) Determine *m\_tuple* of the received MSDU, which is equal to the tuple of all mirror classifier parameters of the MSDU masked by the associated masks (see (b) above). If the MSDU has no value for one or more of the mirror classifier parameters, the remaining steps are not performed.
* (2) Determine *tuple* corresponding to the received MSDU, by mapping all the masked values of parameters in *m\_tuple* per the relationships described in (b) above
* (3) Add or update the variable *UP{tuple}* in the list, setting the value of the variable to the UP of the received MSDU. When a variable is updated (i.e. when a variable corresponding to the same *tuple* already exists in the list), the previous value of the variable is discarded.

If a period equal to the value of the Stream Timeout subfield of the MSCS Descriptor element has elapsed since a given variable in the list was last updated, that variable may be removed from its maintained list.

d) In parallel, classification begins of incoming MSDUs from the DS or WM whose DA parameter value maps to an RA equal to the MAC address of the non-AP STA. The following steps are performed for each such MSDU:

- (1) Determine *tuple* of the MSDU, which is equal to the tuple of all the classifier parameters of the MSDU masked by the associated masks (see (a) above). If the MSDU does not have a value for one or more of the classifier parameters, the remaining steps are not performed

- (2) If variable *UP{tuple}* exists in the list for this value of *tuple*, and the MSDU is not part of a TS (as described in 11.4 (TS operation)), and the MSDU is not part of an SCS stream (see 11.26.2 (SCS Procedures)), assign the MSDU a UP equal to min(*UP{tuple}, UPLim*), where *UPLim* is the value of the User Priority Limit subfield of the MSCS Descriptor element

NOTE – The maintenance of the logical list of variables *UP{tuple}* described above is a logical description of a required process. The specific implementation of this process is vendor specific and is not required to match the description above.

NOTE -- When MSCS does not assign a UP to an MSDU that is not part of a TS or an SCS stream, the UP of the MSDU might be set by other mechanisms such as interworking QoS mapping from DSCP values (e.g. see R.3 (QoS mapping guidelines for interworking with external network)).

NOTE – It is recommended that a non-AP STA using MSCS sets the UP of MSDUs it transmits to its associated AP in a way that does not cause the variables *UP{tuple}* maintained by the AP to change excessively often. This helps to minimize management overhead on the AP, and also minimizes the possibility of MSDUs with the same value of *tuple* being delivered to the higher layers out of order during the transition of the UP of those MSDUs. Note that the requirements for ordering of MSDUs with the same TID value in the MAC sublayer (see 5.1.3 (MSDU ordering)) are unaffected by MSCS.

A non-AP STA may request the termination of a currently active MSCS by sending an MSCS Request frame with the Request Type field set to “Remove” in the MSCS Descriptor element. Upon reception of a request to terminate the active MSCS, the AP shall cease to apply the corresponding classifiers and processing related to the active MSCS and delete the maintained list of variables *UP{tuple}*. The AP shall send an MSCS Response frame to confirm the termination of the MSCS, by including a value of “TCLAS\_PROCESSING\_TERMINATED” in the Status field of an MSCS Response frame and the dialog token in the MSCS Response frame set to the value from the MSCS Request frame that requested termination; an MSCS Descriptor element is not present in the response.

The AP may send an unsolicited MSCS Response frame at any time to cancel an active MSCS, by including a value of “TCLAS\_PROCESSING\_TERMINATED”, “TCLAS\_PROCESSING\_TERMINATED\_INSUFFICIENT\_QOS”, “TCLAS\_PROCESSING\_TERMINATED\_POLICY\_CONFLICT” or “TCLAS\_RESOURCES\_EXHAUSTED” in the Status field of an MSCS Response frame and the dialog token in the MSCS Response frame set to 0. The MSCS Descriptor element is optionally present; if present the Request Type field is set to “Change” and the element indicates a suggested set of parameters that could be accepted by the AP in response to a subsequent request by the recipient non-AP STA.

***Instruct the editor to modify C.3 as follows:***

**C.3 MIB detail**

Dot11StationConfigEntry::=

 SEQUENCE {

......

dot11MSCSActivated TruthValue }

dot11MSCSActivated OBJECT-TYPE

 SYNTAX TruthValue

 MAX-ACCESS read-write

 STATUS current

 DESCRIPTION

"This is a control variable.

It is written by the MAC or an external management entity.

Changes take effect as soon as practical in the implementation.

This variable indicates whether support for Mirrored SCS is enabled on the STA

.

 ::= { dot11StationConfigEntry <ANA>}

DEFVAL { false }