IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Clause 6.3 – Proposed New Text | | | | |
| Date: 2022-07 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Graham SMITH | SR Technology | Sunrise, FL, USA. | 916 799 9563 | [gsmith@srtrl.com](mailto:gsmith@srtrl.com) |
| Mark HAMILTON |  |  |  |  |
| Joseph LEVY | InterDigital | 111 W 33 St, NY, NY, USA | +1 631 622 4139 | jslevy@ieee.org |

Abstract

Proposed Text to replace 6.3

Note: Instructions will also be added to delete 6.3 references from main body. References to the primitives, to be left in. See 21/1822r2.

**6.3 MLME SAP interface**

**6.3.1 Introduction**

The services provided by the MLME to the SME are specified in this subclause. These services are described in an abstract way (following the model described in ITU-T Recommendation X.210 [B55]) and do not imply any particular implementation or exposed interface. MLME SAP primitives are of the general form ACTION.request primitive followed by ACTION.confirm primitive (for an exchange initiated by the SAP client) and ACTION.indication primitive followed by ACTION.response primitive (for an exchange initiated by the MLME). The SME uses the services provided by the MLME through the MLME SAP.

The primitives generally include a STA Address or a Peer STA address. The .confirm primitive generally contains a Result Code. The primitives in 6.3.2 may be used as a guide to the detailed formats of primitives. Parameters that are known to the MLME are not required in the primitive parameters.

**6.3.1.1 Types of MLME-SAP interface primitive forms**

There are six general forms of MLME-SAP interface primitives.

6.3.1.1.1 Type 1

Figure 6.x depicts Type 1. The Type 1 general form is used for the exchange of request/response frames between an initiating STA and a peer STA.

**FIGURE 6.x – Type 1 form of MLME SAP primitives for request/response process**



The .request primitive is generated by the SME of the initiating STA to request that a Request frame is sent to a peer STA.

The .indication primitive is generated by the MLME of the peer STA when the Request frame is received.

The .response primitive is generated by the SME of the peer to request that a Response frame be sent to the initiating STA.

The .confirm primitive includes a Result Code reporting success or failure of the request, and is generated by the MLME of the initiating STA when the either Response frame from the peer STA is acknowledged, or the (re)transmission of the request frame fails.

6.3.1.1.2 Type 2

Figure 6.xx depicts Type 2. The Type 2 general form is used for the SME requesting a process to be initiated by the MLME.

**FIGURE 6.xx – Type 2 form of MLME SAP primitives for SME requesting MLME to perform a process**



The .request primitive is generated by the SME to request that a process is initiated by the MLME.

The .confirm primitive generally includes a Result Code reporting success or failure of the request, and is generated by the MLME when the requested action of process is completed, or fails.

6.3.1.1.3 Type 2a

Figure 6.xxx depicts Type 2a. The Type 2a general form is used for the SME requesting a process to be initiated by the MLME and the SME does not require a confirmation

**FIGURE 6.xxx – Type 2a form of MLME SAP primitives for SME requesting MLME to perform a process not requiring a confirmation**



The .request primitive is generated by the SME to request that a process is initiated by the MLME.

6.3.1.1.4 Type 3

Figure 6.xxxx depicts Type 3. The Type 3 general form is used for the transmission of a frame from one STA to a peer STA that does not require a response from the peer STA but does require a confirmation that the frame was transmitted and either acknowledged or timed out.

**FIGURE 6.xxxx – Type 3 form of MLME SAP primitives for frame transmission not requiring a response, but requiring a confirmation**

The .request primitive is generated by the SME of the initiating STA to request that a Request frame is sent to a peer STA.

The .confirm primitive generally includes a Result Code reporting success or failure of the request, and is generated by the MLME when the requested action of process is completed, or fails.

The .indication primitive is generated by the MLME of the peer STA when the Request frame is received.

6.3.1.1.5 type 4

Figure 6.xxxxx depicts Type 4. The Type 4 general form is used for the transmission of a frame from one STA to a peer STA that does not require a response from the peer STA or a confirmation.

**FIGURE 6.xxxx – Type 4 form of MLME SAP primitives for frame transmission not requiring a response or a confirmation**

The .request primitive is generated by the SME of the initiating STA to request that a Request frame is sent to a peer STA.

The .indication primitive is generated by the MLME of the peer STA when the Request frame is received.

6.3.1.1.6 Type 5

Figure 6.xxxxxx depicts Type 5. The Type 5 general form is used for the transmission of a frame that does not require a response, but does require a confirmation that it was sent.

**FIGURE 6.xxxxxx – Type 5 form of MLME SAP primitives for a frame transmission from a STA, but does require a confirmation that it was sent**



The .request primitive is generated by the SME to request that the MLME transmits a frame.

The .confirm primitive generally includes a Result Code reporting success or failure of the request, and is generated by the MLME when the requested frame transmission is completed, or fails.

6.3.1.1.7 Type 6

Figure 6.xxxxxxx depicts Type 6. The Type 6 general form is used when the MAC informs the SME of an event.

**FIGURE 6.xxxxxxx – Type 6 form of MLME SAP primitives for MAC informing SME of an event**



The .indication primitive is generated by the MLME to inform the SME of an event.

**6.3.2 MLME-SAP Primitives**

MLME-SAP primitives are only detailed in this clause when they do not directly correspond to frame exchanges described in Clause 9, 11, 12, 13 or 14, where the primitive parameters differ significantly from the fields in the respective Action frames, or when the primitives may not be clear from the descriptions in Clauses 9, 11, 12, 13 or 14.

<*Include a sentence here that says something about the extra primitives above the frame exchange frames. i.e. STAAddress or PeerSTAAddress always present and Status in .confirm> OR we add this to the descriptions of the Types, OR is the sentence in 6.3.1 OK?*>

**Include the following in full, renumbering where necessary:**

6.3.2.1 MLME-POWERMGT (6.3.2) Type 2 see 11.2

6.3.2.2 MLME SCAN (6.3.3) Type 2

MLME-SCAN-STOP Type 2a see 6.3.3

6.3.2.3 MLME JOIN (synchronization) (6.3.4) Type 2 see 11.1

6.3.2.4 MLME AUTHENTICATE (6.3.5) Type 1 see 11.3.4

6.3.2.5 MLME DEAUTHENTICATE (6.3.6) Type 3 see 11.3.4

6.3.2.6 MLME ASSOCIATE (6.3.7) Type 1 see 11.3.5

6.3.2.7 MLME REASSOCIATE (6.3.8) Type 1 see 11.3.5

6.3.2.8 MLME DISASSOCIATE(6.3.9) Type 3 see 11.3.5

6.3.2.9 MLME RESET(6.3.10) Type 2a

6.3.2.10 MLME START(6.3.11) Type 2

6.3.2.11 MLME STOP(6.3.12) Type 2a

6.3.2.12 Protocol layer model for spectrum management and radio measurement(6.3.13)

6.3.2.13 MLME SETKEYS (was 6.3.19) Type 2a

6.3.2.14 MLME DELETEKEYS (was 6.3.20) Type 2a

**6.4 Table of MLME SAP interfaces**

Table 6.4 x lists the diagnostic report name, the MLME SAP interface primitives, and the Type as defined in 6.3.1.

.

**Table 6.4.x MLME SA interface**

| **Diagnostic report Name** | **MLME-** | **Type** | **References** | **Comments** |
| --- | --- | --- | --- | --- |
| Power management | POWERMGT | 2 | *6.3.2.1* |  |
| Scan | SCAN | 2 | *6.3.2.2* |  |
| SCAN-STOP | 2a |  |  |
| Synchronization | JOIN | 2 | *6.3.2.3* |  |
| Authenticate | AUTHENTICATE | 1 | *6.3.2.4* |  |
| Deauthenticate | DEAUTHENTICATE | 3 | *6.3.2.5* |  |
| Associate | ASSOCIATE | 1 | *6.3.2.6* |  |
| Reassociate | REASSOCIATE | 1 | *6.3.2.7* |  |
| Disassociate | DISASSOCIATE | 3 | *6.3.2.8* |  |
| Reset | RESET | 2a | *6.3.2.9* |  |
| Start | START | 2 | *6.3.2.10* |  |
| Stop | STOP | 2a | *6.3.2.11* |  |
| Measurement request | MREQUEST | 4 | *6.3.2.12, 9.6.2, 9.6.6* |  |
| Channel measurment | MEASURE | 2 | *6.3.2.12,9.4.2.20, 9.4.2.21* | Contains Measurement Request elements |
| Measurement report | MREPORT | 4 | *6.3.2.12, 9.4.2.21* | Contains Measurement Request elements |
| Channel switch | CHANNELSWITCH | 1 | *6.3.2.12, 9.6.2.6, 9.4.2.18* |  |
| TPC request | TPCADAPT | 5 | *6.3.2.12, 9.6.2.4, 9.6.2.5, 11.7.7* | TPC request and response frames |
| SetKeys | SETKEYS | 2a | *6.3.2.13* |  |
| DeleteKeys | DELETEKEYS | 2a | *6.3.2.14* |  |
| MIC (michael) failure event | MICHAELMICFAILURE | 6 | *12.5.2.4* | Indication only |
| EAPOL | EAPOL | 2 | *12.5.2.4* | Sends an EAPOL frame |
| SetProtection | SETPROTECTION | 2a | *9.2.1.4.9* | Protect descriptors: protect key and key type |
| Protected frame dropped | PROTECTEDFRAMEDROPPED | 6 | *12.6.6* | Used when temporal key unavailable |
| TS management interface | ADDTS | 1 | *9.6.3.3, 9.6.3.2* | See 10.23.3 and 10.23.4 |
|  | DELTS | 6 | *9.6.3.4, 9.4.1.16* |
|  | ADDTSRESERVE | 1 | *9.6.3.7, 9.6.3.8* |
| Higher layer synchronization support | HL-SYNC | 4 | *11.6.2* | Contains only address. Out-of-scope |
| Block Ack | ADDBA | 1 | *9.3.1.7, 9.6.4.2, 9.6.4.3* | See 10.25  Block Ack Parameter Set 9.4.1.13  See DELBA frame parameters 9.4.1.16 |
| DELBA | 4 | *9.6.4.4, 9.4.1.16* |
| Schedule element management | SCHEDULE | 4 | *9.4.2.33* |  |
| Vendor-specific action | VSPECIFIC | 4 |  | Request sending a Vendor Specific frame  Notes: Simply requests sending a Vendor Specific frame – does not specify the frame. |
| Neighbor report | NEIGHBORREPREQ | 4 | *9.6.6.6* | See 11.10.10  Element *9.4.2.36* |
| NEIGHBORREPRESP | 4 | *9.6.6.7* |
| Link Measure Request | LINKMEASUREREQ | 4 | *9.6.6.4* | See 11.10.11  Was Type 5, changed it to 2 type 4s |
| LINKMEASURERES | 4 | *9.6.6.5* |
| Resource request | RESOURCE-REQUEST | 1 | *13.8.4* | See 13.8 |
| RESOURCE-REQUEST-LOCAL | 2 | *13.11.2* |
| Remote request | REMOTE-REQUEST | 4 | *9.6.8* | Sends over the DS requests |
| Extended channel switch announcement | EXTCHANNELSWITCH | 1 | *9.6.7.7,* | Element 9.4.2.52  Response is similar to request but with result code  Note: parameters match. Request sends (Protected) Extended Channel Switch Announcement frame. 9.4.2.52  Response sends “accepted extended channel switch request” same as request frame |
| DSE power constraint announcement | DSETPC | 1 | *9.6.7.10,* | See 4.3.12  Response is similar to request but with result code |
| Enablement | ENABLEMENT | 1 | *9.6.7.4* | Response is similar to Request but with result code  Note: DSE Enablement frame sent. Seems strange?? Just gives results. Same response as request with result code |
| Deenablement | DEENABLEMENT | 4 | *9.6.7.5* | Note: DSE Deenablement frame sent Seems strange?? |
| SA Query support | SA-QUERY | 1 | *9.6.9.2* | Response has same parameters as request |
| Get TSF timer | GETTSFTIME | 2 |  | Request has no parameters. Confirm has TSFtime. |
| Timing Advertisement | TIMING-ADVERTISEMENT | 4 | *9.4.2.60* | See 11.19.2 |
| TDLS Discovery | TDLSDISCOVERY | 1 | *9.6.12.12, 9.6.7.16* | See 11.20.3 |
| TDLS direct-link establishment | TDLSSETUPREQUEST | 4 | *9.6.12.2* | See 11.20 |
| TDLSSETUPRESPONSE | 4 | *9.6.12.3* |
| TDLSCONFIRM | 4 | *9.6.12.4* |
| TDLSPOTENTIALPEERSTA | 2 |  | Request has no parameters. Confirm has RSSI.  NOTE: Can’t find a reference to a (TDLS) potential peer STA in 11.20. |
| TDLS direct link teardown | TDLSTEARDOWN | 4 | *9.6.12.5* | See 11.20 |
| TDLS peer U-APSD | TDLSSPTI | 1 | *9.6.12.6* |
| TDLS channel switch | TDLSCHANNELSWITCH | 1 | *9.6.12.7, 9.6.12.8* |
| TDLS peer PSM | TDLSPEERPSM | 1 | *9.6.12.9, 9.6.12.10* |
| Event | EVLREQUEST | 4 | *9.6.13.2, 9.4.2.66* | Uses event elements 9.4.2.67. Can’t see why this is difference to EVLRequest. It requests MLME to initiate the specified event. |
| EVLREPORT | 4 | *9.6.13.3, 9.4.2.67* |
| EVLOG | 2 | *9.4.2.67* |
| Diagnostic request report | DIAGREQUEST | 4 | *9.6.13.4, 9.4.2.69* |  |
| DIAGREPORT | 4 | *9.6.13.5, 9.4.2.69* |  |
| Location configuration request | LOCATIONCFG | 1 | *9.6.13.6, 9.6.13.7,*  *9.4.2.70* |  |
| Location track notification | LOCATIONTRACKNOTIF | 4 | *9.6.13.6, 9.4.2.70* |  |
| Timing measurement | TIMINGMSMTRQ | 4 | *9.6.13.28* | See 11.21.5 |
| TIMINGMSMT | 3 | *9.6.14.3* |
| Fine timing measurement (FTM) | FINETIMINGMSMTRQ | 4 | *9.6.7.32* | See 11.21.6 |
| FINETIMINGMSMT | 3 | *9.6.7.33* |
| BSS transition management | BTMQUERY | 4 | *9.6.13.8* | See 11.21.7.2 |
| BTM | 1 | *9.6.13.9, 9.6.13.10* | See 11.21.7.3 and 11.21.7.4 |
| FMS setup | FMS | 1 | *9.4.2.75, 9.4.2.76* | See 11.21.8 |
| Collocated Interference request report | CLINTERFERENCEREQUEST | 4 | *9.6.13.13* | See 11.21.9 |
| CLINTERFERENCEREPORT | 4 | *9.6.13.14, 9.4.2.84* |
| TFS setup | TFS | 1 | *9.6.13.15, 9.6.13.16, 9.4.2.79, 9.4.2.80* | See 11.21.12 |
| WNM sleep mode request | SLEEPMODE | 1 | *9.4.2.79, 9.4.2.80* | See 11.2.3.16 |
| TIM broadcast setup | TIMBROADCAST | 1 | *9.4.2.82, 9.4.2.83* | See 11.2.3.15 |
| QoS traffic capability update | QOSTRAFFICCAPUPDATE | 4 | *9.6.13.23,* | See 11.21.10 |
| Channel Usage request | CHANNELUSAGE | 1 | *9.6.13.24, 9.6.13.25, 9.4.2.85,* | See 11.21.15 |
| DMS or GCR request and response procedure | GATS | 1 | *9.6.12.26, 9.4.2.87, 9.6.12.2, 9.4.2.88* | See 11.21.16  . |
| GATS-TERM | 4 | *9.6.12.2, 9.4.2.88* |
| WNM notification request response | WNMNOTIFICATIONREQUEST | 4 | *9.6.13.29* | See 11.21.17  NOTE: Why two type 4’s. Should/could be Type 1? Seems arbitrary what is used. Clould change all Type 1 to 2 type 4’s?? |
|  | WNMNOTIFICATIONRESPONSE | 4 | *9.6.13.30* |
| Network discovery and selection support | GAS | 1 | 9.6.7.12 – 9.6.7.15 | See 11.22.3, 11.23  NOTES:  1) Can be Protected or not, at MLME-user’s request. Do we need an exception to note this control in the parameters?  2) In addition to the peer STAAdress (which can be a broadcast), there is an explicit flag for when this generates a group addressed GAS frame. I don’t understand why that’s needed, so I’m probably missing what it does.  3) The interaction of Comeback Delay and the MLME interface isn’t obvious. But, it appears that the comeback procedure is handled entirely below the MLME SAP, and transparent to the MLME-user. This should probably be noted as an exception. |
| QoS Map element management | QOS-MAP | 4 | 9.6.3.6 | See 11.22.9 |
| Mesh peering management | MESHPEERINGMANAGEMENT | 1 | 9.6.15.2 – 9.6.15.4, | See 14.3, 14.4  NOTE: This is a \_horrible\_ MLME primitive, which just passes the entire contents of up to three different types of frames across the interface. So, basically the MLME-user is the protocol engine, not the MAC. I’m glad to get rid of this one! I’d like to see our structure default to making this primitive instead be a set of primitives for each frame type, with proper parameter lists. |
| Mesh power management | MESHPOWERMGT | 2 |  | See 14.4  NOTE: This looks very similar to 6.3.2 (Power management). No protocol is generated. Do we need another exception for this one? |
| Mesh neighbor offset synchronization | MESHNEIGHBOROFFSETSYNCSTART | 2 |  | See 14.13  NOTE: This feels very similar to 6.3.4 (Synchronization). No protocol is generated. Do we need another exception for this one? |
| MESHNEIGHBOROFFSETCALCULATE | 2 |  | Ditto |
| MESHNEIGHBOROFFSETSYNCSTOP | 2 |  | Ditto |
| Mesh TBTT adjustment | MESHTBTTADJUSTMENT | 1 | 9.6.16.11, 9.6.16.12, | See 14.13.4 |
| MCCA management interface | ACTIVATEMCCA | 2a | 10.24.3.2 | NOTE: Another no protocol one. It seems like an error that this is “2a” and not “2”, since there is another step after a waiting period, and it seems like the .cnf would be a good way to indicate this point. Besides, couldn’t a parameter be wrong, MCCA could already be enabled, etc., just like the MLME-START.confirm ResultCodes? |
| MCCASETUP | 1 | 9.6.16.6, 9.6.16.7, 10.24.3.6 | . |
| MCCAADVERTISEMENT | 1 | 9.6.16.8, 9.6.16.9, 10.24.3.7 | NOTE: The current MLME primitive appears to be incorrect, in missing the “MCCAOP Advertisement Overview” element in the .request, but our scheme will fix that! 😊 |
| MCCATEARDOWN | 4 | 9.6.16.10, 10.24.3.8 |  |
| MBSS congestion control | MBSSCONGESTIONCONTROL | 4 | 9.6.16.5, | See 14.12.2 |
| MBSS proxy update | MBSSPROXYUPDATE | 1 | 9.6.17.2, 9.6.17.3, | See 14.11.4.3 |
| MBSS mesh gate announcement | MBSSGATEANNOUNCEMENT | 4 | 9.6.16.4 | See 14.11.2 |
| Mesh link metric | MESHLINKMETRICREAD | 2 | 14.9.2 | NOTE: This looks very similar to 6.3.40 (Get TSF timer). No protocol is generated |
| MESHLINKMETRICREPORT | 4 | 9.6.16.2, | See 14.8.3, 14.9 |
| HWMP mesh path selection | HWMPMESHPATHSELECTION | 4 | 9.6.16.3, | See 14.10 |
| QMF policy | QMFPOLICY | 4 | 9.6.7.18, | See 11.24.2 |
| QMFPOLICYCHANGE | 4 | 9.6.7.19, |
| QMFPOLICYSET | 2a | 11.24.2.2 | NOTE: This looks very similar to 6.3.19 (SetKeys). No protocol is generated |
| SCS request and response procedure | SCS | 1 | 9.6.18.2, 9.6.18.3, | See 11.25.2. |
| SCS-TERM | 4 | 9.6.18.3, | See 11.25.2  NOTE: The procedure in clause 11 is not explicitly connected to this SAP primitive, and it has no unique protocol (frame). So, it will be lost with our scheme. Maybe that’s okay?? |
| QLoad report management | QLOAD | 1 | 9.6.7.20, 9.6.7.21, | See 11.26.2 |
| HCCA TXOP advertisement management | TXOPADVERTISEMENT | 1 | 9.6.7.22, 9.6.7.23, | See 11.26.3 |
| GCR group membership management | GROUP-MEMBERSHIP | 1 | 9.6.18.4, 9.6.18.5, | See 11.21.16.3.2 |
| AP PeerKey management | APPEERKEY | 4 | 9.6.7.24, | See 12.10.2 |
| On-channel Tunneling operation | OCTunnel | 6 | 9.6.7.47, 9.6.20.7, | See 11.31.5 |
| Multi-band operation | FST-SETUP | 1 | 9.6.20.2, 9.6.20.3, | See 11.31.3  NOTE: Like Mesh Peering Management, this is a \_horrible\_ MLME primitive. Glad we’re cleaning it up. |
| FST-ACK | 1 | 9.6.20.5, 9.6.20.6, | See 11.31.3 |
| FST-TEARDOWN | 4 | 9.6.20.4 | See 11.31.4 |
| FST-INCOMING | 2a | 11.31.3.2 |  |
| DMG relay operation | RELAY-SEARCH | 1 | 9.6.19.8, 9.6.19.9, | See 11.34.2  NOTE: No good analogy for this one in the other “2a”s, but probably handle somewhat similarly to those. |
| RLS | 1 | 9.6.19.12, 9.6.19.13, | See 11.34.2.4 |
| RLS-TEARDOWN | 4 | 9.6.19.15, | See 11.34.4 |
| Quieting adjacent BSS operation | QAB | 1 | 9.6.7.34, 9.6.7.35 | See 11.35 |
| DMG beamforming | BF-TRAINING | 3 | 10.42 | NOTE: This is all very like SCAN. Kicking off these primtives starts a whole bunch of under-the-hood protocol stuff, which eventually gets reported as completed with some result. Probably need to keep this as an explicit case. |
| SU-MIMO-BF-TRAINING | 3 |  | See above |
| MLME-MU-MIMO-BF-TRAINING | 3 |  |  |
| SU-MIMO-HYBRID-BF-PROTOCOL | 3 |  |  |
| MU-MIMO-HYBRID-BF-PROTOCOL | 3 |  |  |
| PN event report | PN-EXHAUSTION | 6 |  | 12  NOTE: No protocol, local indication only. This occurs in multiple places in clause 12 (for different security models). No good single reference. Probably need to keep this as an explicit case. |
| PN-WARNING | 6 |  | See above |
| Channel Availability Query | CHANNELAVAILABILITYQUERY | 1 | 9.6.7.25 | See 11.42.4  NOTE: Another \_horrible\_ MLME primitive. Glad we’re cleaning it up. |
| Channel schedule management | CHANNELSCHEDULEMANAGEMENT | 1 | 9.6.7.26 | See 11.42.5 |
| Contact verification signal | CVS | 4 | 9.6.7.27 | See 11.42.6 |
| GDD Enablement | GDDENABLEMENT | 1 | 9.6.7.28, 9.6.7.29 | See 11.42.2, 11.42.3 |
| Network channel control management | NETWORKCHANNELCONTROL | 1 | 9.6.7.30 | See 11.42.7 |
| White space map (WSM) | WSM | 4 | 9.6.7.31, | See 11.42.8 |
| Estimated Throughput | ESTIMATED-THROUGHPUT | 2 | 11.44 | NOTE: No protocol, like mesh link metric. |
| Get authentication and association state | GETAUTHASSOCSTATE | 2 | 11.3.1 | NOTE: Who the heck thought this was necessary?? |
| FILS Container | FILSContainer | 1 | 9.4.2.184 |  |
| Dynamic AID assignment operation | AIDSWITCH | 1 | 9.6.24.2  9.6.24.3 | AID Switch Request  AID Switch Response |
| Sync Control | SYNCCONTROL | 4 | 9.6.24.4 |  |
| STA Information Announcement | STAINFORMATION | 4 | 9.6.24.5 | STA Information Announcement |
| EDCA Parameter Set update | EDCAPARAMETERSET | 4 | 9.6.24.6 |  |
| EL Operation | ELOPERATION | 4 | 9.6.24.7 |  |
| TWT Setup | TWTSETUP | 1 | 10.47.1  9.6.24.8 |  |
| TWT Teardown | TWTTEARDOWN | 4 | 9.6.24.9 |  |
| Sectorized Group ID List management | SECTORIZEDGROUPID | 4 | 9.6.24.10 |  |
| Header Compression procedure | HEADERCOMPRESSION | 1 | 9.6.25.5 | Header Compression frame |
| Reachable Address Update | REACHABLEADDRESSUPDATE | 4 | 9.6.25.2 |  |
| Control response MCS negotiation operation | CONTROLRESPONSEMCS | 1 | 9.6.27.2  9.6.27.3 | Control Response MCS Negotiation Request  Control Response MCS Negotiation Response |
| S1G relay (de)activation | S1GRELAYACTIVATE | 1 | 9.6.25.3  9.6.25.4 | Relay Activation Request  Relay Activation Response |
| DCS procedure | DCSMEASUREMENT | 1 | 9.6.7.37  9.6.7.38 | DCS Measurement Request  DCS Measurement Response |
| Update | UPDATE | 2 |  | ServiceHint and ServiceHash |
| MSCS request and response procedure | MSCS | 1 | 9.6.18.6  9.6.18.7 | MSCS Request  MSCS Response |
| MSCS-TERM | 4 | 9.6.18.7 | MSCS Response to terminate MSCS |
| MAC Address Update | UPDATEMACADDRESS | 2 | 12.2.10 | MAC address change required |
| Quiet time period | QTP | 1 | 26.17.5.2  26.17.5.3  9.6.31.3 | QTP Request  QTP Response  QTP Action frame |
| TDD beamforming | TDD-BF-TRAINING | 3 | 10.42 | TDD beamforming training procedures |
| TDD sector switch | TDD-SECTOR-SWITCH | 3 | 11.36.3 |  |
| TDD beam measurement | TDD-BEAM-MEASUREMENT | 3 | 10.42.11  11.36.4 | TDD beamforming  TDD beam measurment |
| TDD structure and schedule | TDD-SLOT-STRUCTURE | 2 | 9.4.2.281 | See 10.39.6.2.2  See 11.54 |
| TDD-SLOT-SCHEDULE | 2 | 9.4.2.282 |
| TDD-SLOT-ANNOUNCE | 3 | 9.4.2.281  9.4.2.282 |
| TDD-BANDWIDTH | 3 | 9.4.2.285 |
| WUR mode setup | WURMODESETUP | 1 | 9.6.33.2 | See 29.8.2, |
| WUR mode teardown | WURMODETEARDOWN | 4 | 9.6.33.3 | See 29.8.2 |
| WUR Discovery | WURDISCOVERY | 2 | 9.9.3.3 | See 29.12 |