RW-WLAN-nX LMAC SW

User Manual

RW-WLAN-nX-LMAC-SW-UM/1.11

Version 1.11

2016-08-02

RivieraWaves
http://www.rivierawaves.com/

Revision History

Version	Date	Revision Description	Author
0.01	2012-03-30	Initial Release	Steven L'Her
0.02	2012-04-05	Added debug commands	Steven L'Her
0.03	2012-04-06	Added parameters of the STA ADD/DEL commands	Steven L'Her
0.04	2012-05-11	Updated MM_VERSION_CFM parameters	Steven L'Her
0.05	2012-05-31	Added API usage section Removed MM_SET_POWERMODE_REQ/CFM Added MM_SET_IDLE_REQ/CFM Added MM_BA_ADD_REQ/CFM Added MM_BA_DEL_REQ/CFM Added MM_PRIMARY_TBTT_IND Added MM_SECONDARY_TBTT_IND	Steven L'Her
0.06	2012-06-06	Added interface number parameter to the MM_STA_ADD_REQ	Steven L'Her
0.07	2012-06-27	Added Aggregation related Transmit Descriptor parameters and BA agreement management explanations	Cristina Ionescu
0.08	2012-08-02	Added RX description	Steven L'Her
1.0	2012-12-21	Minor edits	Steven L'Her
1.01	2013-01-28	RX and TX descriptor updates	Steven L'Her
1.02	2013-05-22	Set channel API changes	Steven L'Her
1.03	2013-09-25	Added MM_SET_POWER_REQ/CFM. Updated MM_START_REQ/MM_BA_ADD_REQ message parameters.	Steven L'Her
1.04	2014-02-11	Updated TX API. Added power-save and connection monitoring APIs.	Steven L'Her
1.05	2015-06-24	Updated MM API	Steven L'Her
1.06	2015-09-23	Updated MM API with messages dedicated for channel context management	Laurent Trarieux
1.07	2016-03-31	Added MM_BFMER_ENABLE_REQ, MM_CHANNEL_SURVEY_IND and MM_CSA_COUNTER_IND messages	Laurent Trarieux
1.08	2016-03-31	Added missing MM message descriptions. Updated message parameter structures.	Steven L'Her
1.09	2016-04-12	Add messages for P2P PS mode management	Laurent Trarieux
1.10	2016-05-04	Added description of Linux commands allowing to control the P2P PS Mode	Laurent Trarieux
1.11	2016-08-02	Added MU-MIMO message descriptions	Steven L'Her

Changes between a version and the previous one is reflected by the addition of **change bars**, like for the line below:

TBD	Date entered	Description	Status

Items to be determined in the future versions of this document

Table of Contents

List of Figures

List of Tables

1 Overview

1.1 Document overview

1.1.1 Purpose

The purpose of this document is to describe the functionality and the API of the RW-WLAN-nX LMAC SW, as well as guidelines of usage of this API to put the system in the required mode. This document is addressed to both upper MAC developers and any person wishing to have an overview of the LMAC functionality.

1.1.2 Abbreviations and Acronyms

AC	Access Category
ACK	Acknowledgment
ACM	Access Category Mandatory
ACS	Automatic Channel Selection
AID	Association Identifier
AP	Access Point
APSD	Automatic Power Save Delivery
A-MPDU	Aggregate MAC Protocol Data Unit
A-MSDU	Aggregate MAC Service Data Unit
BA	Block Acknowledgement
BAR	Block Acknowledgement Request
BFMER	Beamformer
BSSID	Basic Service Set Identifier
CF	Contention Free
CSA	Channel Switch Announcement
CSI	Carrier State Information
CTS	Clear To Send
CW	Contention Window
DCF	Distributed Coordination Function
EDCA	Enhanced Distributed Channel Access
FIFO	First-In-First-Out
FC	Frame Control
FCS	Frame Check Sequence
HCCA	HCF Controlled Channel Access
HT	High Throughput
IBSS	Independent Basic Service Set
ICV	Integrity Check Value
IV	Initialization Vector
LLC	Logical Link Control

L-SIG	Legacy (Non-HT) Signal Field
MAC	Medium Access Control
MEM-BAR	Memory Base Address Register
MIB	Management Information Base
MIMO	Multiple Input Multiple Output
MLME	MAC Sub Layer Management Entity
MMPDU	MAC Management Protocol Data Unit
MPDU	MAC Protocol Data Unit
MSDU	MAC service data unit
NAV	Network Allocation Vector
NoA	Notice of Absence
OS	Operating System
OppPS	Opportunistic Power Save
P2P	Peer-to-Peer
PC	Point Coordinator
PHY	Physical layer
PLME	PHY Sub Layer Management Entity
PS	Power Save
PSMP	Power Save Multi-Poll
QAP	QoS Access Point
QoS	Quality of Service
QSTA	QoS Station
RD	Reverse Direction
RDG	Reverse Direction Grant
RoC	Remain on Channel
RSNA	Robust Security Network Association
RTS	Request to Send
RX	Receiver
SSID	Service Set Identifier
STA	Station
STBC	Space-Time Block Coding
TID	Traffic Identifier
TIM	Traffic Indication Map
TS	Traffic Stream
TU	Time Unit
TX	Transmitter

TX_REQ	Transmit Request message received from UMAC-SW
U-APSD	Unscheduled Automatic Power Save Delivery
WEP	Wired Equivalent Privacy
WLAN	Wireless LAN
WM	Wireless Medium

Table 1: Abbreviations and Acronyms

2 LMAC SW Overview

This section describes major components of LMAC SW and its interfaces with other layers in the embedded system. The figure below shows the different blocks of the LMAC SW.

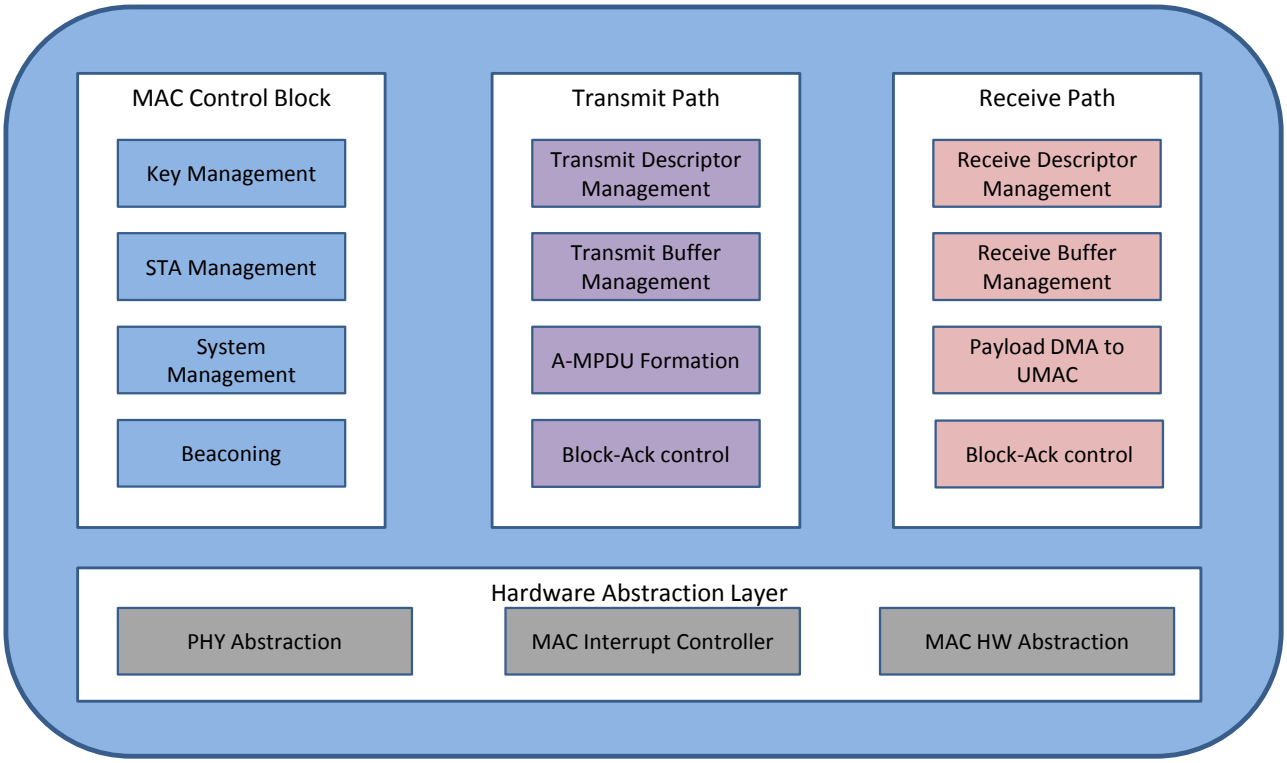


Figure 1: LMAC SW in split MAC SW partitioning mode

Following sections briefly describe each of the layers present in the embedded system.

2.1 Aggregation

Aggregation is one of the main features added by the 11n section.

This feature allows two devices to establish an agreement under which the Originator may send multiple MPDUs in an A-MPDU without waiting for an ACK frame for each, but for a Block Ack frame issued by the Responder of this agreement.

Of course, for such an agreement to be possible, several Block Ack features of the two devices must be compatible. The devices present their features under the BA establishment and if successful, they use some resulting parameters for the data transfers (e.g.: maximum number of MPDUs in an AMPDU, maximum length of an AMPDU, minimum MPDU start spacing interval, Start Sequence Number, etc.)

In order to implement aggregation in the nX system and in LMAC FW more particularly, several decisions have been taken

- Due to mac80211 implementation
  - *only A-MPDUs* will be implemented (not A-MSDUs also)
  - under *HT-Immediate Block Ack* Agreements (hard coded in mac80211)
- All aggregation related additions to LMAC FW are to be kept under a *compilation flag* (AGG=on/off), allowing the FW to run identically with/without aggregation for a singleton MPDU. **However,** several interface parameters are added with aggregation feature enabled, so the UMAC interface to LMAC must be adapted the same way.

- Due to system capabilities, Minimum MPDU Start Spacing time may be chosen internally as other microseconds multiples than [1/4, ½, 1, 2, 4, 8, 16]
- Reception of MPDUs of an AMPDU is quite invisible to LMAC, HW handles deaggregation and there is only one bit in the Reception Header Descriptors which informs that the MPDU was part of an AMPDU. Mac80211 handles reordering.

The extra parameters added to the Transmission Header Descriptor are described in the Data transmission chapter.

The extra API messages are described in the API chapter.

The confirmation status interpretation for aggregated MPDUs changes slightly and is described in the… chapter.

3 API description

3.1 Configuration interface

This interface is used for the configuration of the LMAC SW. The parameters that can be configured include:

- The channel the PHY is tuned to
- The BSS parameters (BSSID, Basic Rates, EDCA, etc.)
- The state of the system (stopped, started, active, etc.)
- The receive filters
- The known stations and encryption keys

The configuration interface is also used to control the debug features of the LMAC:

- Enabling/Disabling trace functions
- Reading/writing memory

This interface is controlled via messages that are sent/received to/from the LMAC SW. These messages have to be addressed to the correct module of the LMAC, using a unique 16-bit identifier (see Table 2).

LMAC Module	Identifier
MAC Management (MM)	0x0000
Debug Configuration (DBG)	0x0001

Table 2: LMAC module identifiers

A naming convention is adopted, in order to easily differentiate the direction of each message:

- xxxx_REQ: Configuration request. It is sent from upper MAC to LMAC.
- xxxx_CFM: Confirmation of a previous configuration request. It is sent from LMAC to upper MAC.
- yyyy_IND: Asynchronous indication sent from LMAC to upper MAC.

The general format of such messages is given in the below section.

3.1.1 Control Message format

A control message has to follow the following format:

name	msg_id	dest_id	src_id	param_len	parameters
size	2	2	2	2	0 - n

Figure 2: Control Message Format

The fields of a control message are the following:

- msg_id: Unique identifier of the message (see below chapters for details about this field)
- dest_id: Destination of the message (see Table 2)
- src_id: Reserved for internal LMAC use. Has to be put to 0x64 in all messages.
- param_len: Length, in bytes, of the parameters field (0 if no parameters)
- parameters: Parameters of the message, if any. This field varies depending on the value of msg_id.

3.1.2 MAC Management API description

3.1.2.1 Overview

Request	Confirmation	Description
MM_RESET_REQ	MM_RESET_CFM	Put back the LMAC SW, as well as MAC and PHY HW to their default state
MM_START_REQ	MM_START_CFM	Start the operation of the LMAC SW, MAC HW and PHY in monitor mode on default channel
MM_VERSION_REQ	MM_VERSION_CFM	Read the version of the LMAC SW, MAC HW and PHY
MM_ADD_IF_REQ	MM_ADD_IF_CFM	Add an interface (e.g. of type STA, AP, etc.). At least one interface has to be added for operation other than monitor mode.
MM_REMOVE_IF_REQ	MM_REMOVE_IF_CFM	Remove an interface previously added
MM_STA_ADD_REQ	MM_STA_ADD_CFM	Add a peer STA to the LMAC
MM_STA_DEL_REQ	MM_STA_DEL_CFM	Delete a peer STA from the LMAC
MM_SET_FILTER_REQ	MM_SET_FILTER_CFM	Set the receive filter of the MAC HW
MM_SET_CHANNEL_REQ	MM_SET_CHANNEL_CFM	Tune the PHY to the desired channel, band and type (i.e. 20/40/80MHz)
MM_SET_DTIM_REQ	MM_SET_DTIM_CFM	Set the DTIM value to the MAC HW
MM_SET_BEACON_INT_REQ	MM_SET_BEACON_INT_CFM	Set the beacon interval to the MAC HW
MM_SET_BASIC_RATES_REQ	MM_SET_BASIC_RATES_CFM	Set the basic rates to the MAC HW
MM_SET_BSSID_REQ	MM_SET_BSSID_CFM	Set the BSSID to the MAC HW
MM_SET_EDCA_REQ	MM_SET_EDCA_CFM	Set the EDCA TX parameters to the MAC HW
MM_SET_MODE_REQ	MM_SET_MODE_CFM	Set the ABGNAC mode to the MAC HW
MM_SET_VIF_STATE_REQ	MM_SET_VIF_STATE_CFM	Set the state of the VIF upon an association/AP starting
MM_SET_SLOTTIME_REQ	MM_SET_SLOTTIME_CFM	Set the slot duration to the MAC HW
MM_SET_IDLE_REQ	MM_SET_IDLE_CFM	Put MAC HW into/out of IDLE state
MM_KEY_ADD_REQ	MM_KEY_ADD_CFM	Add an encryption key to the LMAC SW/MAC HW
MM_KEY_DEL_REQ	MM_KEY_DEL_CFM	Delete an encryption key previously added
MM_BA_ADD_REQ	MM_BA_ADD_CFM	Add a Block Ack agreement parameters to the attached STAID in STA INFO TABLE, by TID.
MM_BA_DEL_REQ	MM_BA_DEL_CFM	Delete a Block Ack agreement parameters from the attached STAID in STA INFO TABLE, by TID.
MM_SET_POWER_REQ	MM_SET_POWER_CFM	Sets the TX power that has to be used for HW and LMAC generated frames
MM_DBG_TRIGGER_REQ	n/a	Forces the LMAC to trigger the embedded logic analyser (if available), and forward a debug dump to the Upper MAC.

MM_SET_PS_MODE_REQ	MM_SET_PS_MODE_CFM	Sets the Power-Save mode state.
MM_CHAN_CTXT_ADD_REQ	MM_CHAN_CTXT_ADD_CFM	Add a channel context.
MM_CHAN_CTXT_DEL_REQ	MM_CHAN_CTXT_DEL_CFM	Delete a channel context
MM_CHAN_CTXT_LINK_REQ	MM_CHAN_CTXT_LINK_CFM	Link a VIF with a channel context
MM_CHAN_CTXT_UNLINK_REQ	MM_CHAN_CTXT_UNLINK_CFM	Unlink a VIF from its channel context
MM_CHAN_CTXT_UPDATE_REQ	MM_CHAN_CTXT_UPDATE_CFM	Update channel context parameters
MM_CHAN_CTXT_SCHED_REQ	MM_CHAN_CTXT_SCHED_CFM	Request to switch to a channel context in order to transmit a management frame.
MM_BCN_CHANGE_REQ	MM_BCN_CHANGE_CFM	Request to change the beacon template in LMAC
MM_TIM_UPDATE_REQ	MM_TIM_UPDATE_CFM	Request to update the TIM in the beacon (i.e to indicate traffic bufferized at AP)
MM_REMAIN_ON_CHANNEL_REQ	MM_REMAIN_ON_CHANNEL_CFM	Request presence on a channel for a given duration
MM_SET_PS_OPTIONS_REQ	MM_SET_PS_OPTIONS_CFM	Sets the Power-Save options (e.g. listen interval).
MM_BFMER_ENABLE_REQ	n/a	Enable support of the Beamformee feature after connection with a Beamformee Capable device.
MM_SET_P2P_NOA_REQ	MM_SET_P2P_NOA_CFM	Request to start/stop a P2P Notice of Absence (NoA) scheduling – For GO interfaces only
MM_SET_P2P_OPPPS_REQ	MM_SET_P2P_OPPPS_CFM	Request to start/stop use of P2P Opportunistic Power Save Mode (OppPS) – For GO interfaces only
MM_CFG_RSSI_REQ	n/a	Request to setup a RSSI threshold
MM_MU_GROUP_UPDATE_REQ	MM_MU_GROUP_UPDATE_CFM	Request to indicate to the FW that a given station now belongs to some MU groups.

Table 3: MAC Management Control messages

A few indications can also be received asynchronously from the LMAC:

Indication	Description
MM_PRIMARY_TBTT_IND	Indicates the primary TBTT to the upper MAC. Upon the reception of this message the upper MAC has to push the beacon(s) to the beacon transmission queue.
MM_SECONDARY_TBTT_IND	Indicates the secondary TBTT to the upper MAC. Upon the reception of this message the upper MAC has to push the secondary beacon(s) to the beacon transmission queue.
MM_CONNECTION_LOSS_IND	Indicates the loss of the connection with the Access Point.
MM_CHANNEL_SWITCH_IND	Indicate that LMAC has switched to a new channel
MM_CHANNEL_PRE_SWITCH_IND	Indicate that LMAC is about to switch on a new channel
MM_REMAIN_ON_CHANNEL_EXP_IND	Indicate the end of the Remain on Channel duration
MM_PS_CHANGE_IND	Indicate a PS state change of a peer device

MM_TRAFFIC_REQ_IND	Indicate that some buffered traffic should be sent to the peer device
MM_P2P_VIF_PS_CHANGE_IND	Indicate a PS state update for a given P2P VIF
MM_CHANNEL_SURVEY_IND	Indicate sensing information for a given channel
MM_CSA_COUNTER_IND	Indicate that CSA counter has been updated
MM_P2P_NOA_UPD_IND	Indicate that a new NoA scheduling has been started or stopped
MM_RSSI_STATUS_IND	Indicate that the RSSI has crossed the defined RSSI threshold
MM_CSA_FINISH_IND	Indicate the completion of the CSA procedure
MM_CSA_TRAFFIC_IND	Indicate that traffic needs to be stopped/restarted depending on the progress of the CSA procedure

Table 4: MAC Management Indication messages

3.1.2.2 Detailed description

3.1.2.2.1 MM_RESET_REQ

This request puts back the LMAC SW, as well as the MAC HW and PHY to their default configuration. After this request is executed, the system is in IDLE state, waiting for configuration by the upper MAC.

Msg_id	0x0000
Confirmation message	MM_RESET_CFM

Table 5: MM_RESET_REQ description

3.1.2.2.2 MM_RESET_CFM

This message confirms the completion of the handling of MM_RESET_REQ message.

Msg_id	0x0001
Confirmation message	n/a

Table 6: MM_RESET_CFM description

3.1.2.2.3 MM_START_REQ

This request starts the operation of the LMAC SW, MAC HW and PHY in monitor mode. Upon reception of this message, the LMAC starts the RF on the default channel and enables the MAC HW for reception of all types of frames, including erroneous frames. No MAC address is set to MAC HW at this point, so automatic response mechanism is disabled (i.e. no ACK, no CTS sent). This mode can be used as a sniffer mode.

It is only handled by the LMAC when in IDLE state (i.e. after initialization of the LMAC, or after handling of a MM_RESET_REQ).

msg_id	0x0002
Confirmation message	MM_START_CFM

Table 7: MM_START_REQ description

3.1.2.2.3.1 Parameters

Name	Type	Description
phy_cfg	struct phy_cfg_tag	Parameters required for the PHY initialization (used radio paths, other cfg). The detailed format of this parameter is dependent on the PHY driver.
uapsd_timeout	u32	Periodicity of transmission of the U-APSD trigger frames if no TX traffic is ongoing (in ms).
lp_clk_accuracy	u16	Accuracy of the low power clock (in ppm)
noa_abs_dur	u8	P2P NoA Absence Duration (in ms) – meaningless if P2P_GO_PS not supported
oppps_ctw	u8	P2P Opportunistic Power Save - CTWindow duration when GO (in ms) - meainingless if P2P_GO_PS not supported

Table 8: MM_START_REQ parameters

3.1.2.2.4 MM_START_CFM

This message confirms the completion of the handling of MM_START_REQ message.

Msg_id	0x0003
Confirmation message	n/a

Table 9: MM_START_CFM description

3.1.2.2.5 MM_VERSION_REQ

This message allows getting the LMAC SW, MAC HW and PHY version.

Msg_id	0x0004
Confirmation message	MM_VERSION_CFM

Table 10: MM_VERSION_REQ description

3.1.2.2.6 MM_VERSION_CFM

This message sends back the version information, following the reception of a MM_VERSION_REQ message.

Msg_id	0x0005
Confirmation message	n/a

Table 11: MM_VERSION_CFM description

3.1.2.2.6.1 Parameters

Name	Type	Description
version_lmac	u32	LMAC version. The formatting of this field is done as follows: - Bits 31-24: Major number - Bits 23-16: Minor number - Bits 15-8: Build number - Bits 7-0: Patch number
version_machw_1	u32	MAC HW version 1 (as formatted in version1Reg MAC register)
version_machw_2	u32	MAC HW version 2 (as formatted in version2Reg MAC register)
version_phy_1	u32	PHY version 1 (formatting is PHY dependent)
version_phy_2	u32	PHY version 2 (formatting is PHY dependent)

Table 12: MM_VERSION_CFM parameters

3.1.2.2.7 MM_ADD_IF_REQ

This request allows leaving the monitor mode to enter the normal operation mode. Once it has been handled, the system has a MAC address and sends control response frames when it receives packets having this MAC address as ADDR1. The receive filter is also reconfigured upon reception of this message, so that only valid frames are forwarded to the upper MAC.

It is not handled by the LMAC when in IDLE state (i.e. after initialization of the LMAC, or after handling of a MM_RESET_REQ), and therefore a MM_START_REQ message has to be issued prior to the MM_ADD_IF_REQ.

msg_id	0x0006
Confirmation message	MM_ADD_IF_CFM

Table 13: MM_ADD_IF_REQ description

3.1.2.2.7.1 Parameters

Name	Type	Description
type	u8	Type of the interface to be added (0x00: STA, 0x01: IBSS, 0x02: AP)
addr	u8[6]	MAC address of the interface

p2p	u8	Flag indicating if the interface of type STA or AP is actually supposed to be a P2P-Client or P2P-GO. 0: Not a P2P interface 1: P2P Client or GO depending on the type parameter (STA or AP)

Table 14: MM_ADD_IF_REQ parameters

3.1.2.2.8 MM_ADD_IF_CFM

This message confirms the completion of the handling of MM_ADD_IF_REQ message. One of the parameters of this confirmation is the instance number of the new interface. This instance number has to be stored by the upper MAC, in order to be used later in some other MM requests.

Msg_id	0x0007
Confirmation message	n/a

Table 15: MM_ADD_IF_CFM description

3.1.2.2.8.1 Parameters

Name	Type	Description
status	u8	Status of the interface adding procedure (0x00 if successful, different from 0 otherwise)
inst_nbr	u8	Instance number of the interface newly added

Table 16: MM_ADD_IF_CFM parameters

3.1.2.2.9 MM_REMOVE_IF_REQ

This request allows removing an interface previously added. The MAC address associated to this interface is removed and the MAC HW will therefore not send control response frame when it receives frames with this MAC address as ADDR1.

At least one MM_ADD_IF_REQ message has to be transmitted prior to issue MM_REMOVE_IF_REQ.

msg_id	0x0008
Confirmation message	MM_REMOVE_IF_CFM

Table 17: MM_REMOVE_IF_REQ description

3.1.2.2.9.1 Parameters

Name	Type	Description

inst_nbr	u8	Instance number of the interface to be removed

Table 18: MM_REMOVE_IF_REQ parameters

3.1.2.2.10 MM_REMOVE_IF_CFM

This message confirms the completion of the handling of MM_REMOVE_IF_REQ message.

Msg_id	0x0009
Confirmation message	n/a

Table 19: MM_REMOVE_IF_CFM description

3.1.2.2.11 MM_STA_ADD_REQ

This request is used to add a station to the list of stations known by the LMAC. It is useful to indicate to the LMAC some information about this station such as the aggregation parameters.

Msg_id	0x000A
Confirmation message	MM_STA_ADD_CFM

Table 20: MM_STA_ADD_REQ description

3.1.2.2.11.1 Parameters

Name	Type	Description
ampdu_size_max_vht	u32	Maximum size (in bytes) of the VHT A-MPDUs the peer STA can support in reception.
ampdu_size_max_ht	u16	Maximum size (in bytes) of the HT A-MPDUs the peer STA can support in reception.
mac_addr	u8[6]	MAC address of the peer STA
ampdu_spacing_min	u8	Minimal spacing between 2 MPDUs in an A-MPDU sent to the peer STA. Expressed in us.
inst_nbr	u8	Index of the virtual interface the peer STA is attached to, as returned in MM_ADD_IF_CFM message.

Table 21: MM_STA_ADD_REQ parameters

3.1.2.2.12 MM_STA_ADD_CFM

This message confirms the completion of the handling of MM_STA_ADD_REQ message. The parameters of this confirmation contain the identifiers of the new station. These identifiers have to be stored by the upper MAC, in order to be used later in some other MM requests or DATA transmission.

msg_id	0x000B
Confirmation message	n/a

Table 22: MM_STA_ADD_CFM description

3.1.2.2.12.1 Parameters

Name	Type	Description
status	u8	Status of the station adding procedure (0x00 if successful, different from 0 otherwise)
sta_idx	u8	Index of the new station
hw_sta_idx	u8	Index of the new station in the MAC HW key storage

Table 23: MM_STA_ADD_CFM parameters

3.1.2.2.13 MM_STA_DEL_REQ

This request is used to remove a station from the list of known stations.

msg_id	0x000C
Confirmation message	MM_STA_DEL_CFM

Table 24: MM_STA_DEL_REQ description

3.1.2.2.13.1 Parameters

Name	Type	Description
sta_idx	u8	Index of the station to be removed. This index is the one provided in the sta_idx parameter of the MM_STA_ADD_CFM message.

Table 25: MM_STA_DEL_REQ parameters

3.1.2.2.14 MM_STA_DEL_CFM

This message confirms the completion of the handling of MM_STA_DEL_REQ message.

msg_id	0x000D
Confirmation message	n/a

Table 26: MM_STA_DEL_CFM description

3.1.2.2.15 MM_SET_FILTER_REQ

This request is used to set the receive filter of the MAC HW. Depending on the state of the system (monitoring, scanning, connection), it allows the upper MAC to minimize the receptions of useless frames.

msg_id	0x000E
Confirmation message	MM_SET_FILTER_CFM

Table 27: MM_SET_FILTER_REQ description

3.1.2.2.15.1 Parameters

Name	Type	Description
filter	u32	RX filter to be put into rxCntrlReg MAC HW register. See [3] for details about this register.

Table 28: MM_SET_FILTER_REQ parameters

3.1.2.2.16 MM_SET_FILTER_CFM

This message confirms the completion of the handling of MM_SET_FILTER_REQ message.

msg_id	0x000F
Confirmation message	n/a

Table 29: MM_SET_FILTER_CFM description

3.1.2.2.17 MM_SET_CHANNEL_REQ

This request is used to tune the PHY to the desired channel. Parameters of this request include the band on which we have to tune to (2.4GHz or 5GHz, and the type of channel, i.e. 20/40/80/160MHz).

msg_id	0x0010
Confirmation message	MM_SET_CHANNEL_CFM

Table 30: MM_SET_CHANNEL_REQ description

3.1.2.2.17.1 Parameters

Name	Type	Description
band	u8	0x00: 2.4GHz 0x01: 5GHz Only one of the above value may be valid depending on the used PHY (e.g. only supporting 2.4GHz band).

type	u8	Type of the channel to be used (e.g. bandwidth) 0x00: 20MHz 0x01: 40MHz 0x02: 80MHz 0x03: 160MHz 0x04: 80+80MHz Some of the above values may not be valid depending on the used PHY (e.g. only supporting up to 40MHz bandwidth).
prim20_freq	u16	Center frequency (in MHz) of Primary 20MHz channel.
center1_freq	u16	Center frequency (in MHz) of the 20, 40 or 80 MHz channel used, or of the Primary 80+80 channel.
center2_freq	u16	Center frequency (in MHz) of the non-contiguous secondary 80+80.
index	u8	Index of the RF for which the channel has to be set: 0: operating (primary) 1: secondary RF (used for additional radar detection). This parameter is reserved if no secondary RF is available in the system
tx_power	s8	Maximum TX power allowed for this channel (in dBm)

Table 31: MM_SET_CHANNEL_REQ parameters

3.1.2.2.18 MM_SET_CHANNEL_CFM

This message confirms the completion of the handling of MM_SET_CHANNEL_REQ message.

msg_id	0x0011
Confirmation message	n/a

Table 32: MM_SET_CHANNEL_CFM description

3.1.2.2.19 MM_SET_DTIM_REQ

This request is used to set the DTIM period parameter to the MAC HW.

msg_id	0x0012
Confirmation message	MM_SET_DTIM_CFM

Table 33: MM_SET_DTIM_REQ description

3.1.2.2.19.1 Parameters

Name	Type	Description

dtim_period	u8	DTIM period of the BSS.

Table 34: MM_SET_DTIM_REQ parameters

3.1.2.2.20 MM_SET_DTIM_CFM

This message confirms the completion of the handling of MM_SET_DTIM_REQ message.

msg_id	0x0013
Confirmation message	n/a

Table 35: MM_SET_DTIM_CFM description

3.1.2.2.21 MM_SET_BEACON_INT_REQ

This request is used to set the beacon interval to the MAC HW.

msg_id	0x0014
Confirmation message	MM_SET_BEACON_INT_CFM

Table 36: MM_SET_BEACON_INT_REQ description

3.1.2.2.21.1 Parameters

Name	Type	Description
beacon_int	u16	Beacon interval of the BSS, in TUs.
inst_nbr	u8	Index of the interface for which the parameter is configured

Table 37: MM_SET_BEACON_INT_REQ parameters

3.1.2.2.22 MM_SET_BEACON_INT_CFM

This message confirms the completion of the handling of MM_SET_BEACON_INT_REQ message.

msg_id	0x0015
Confirmation message	n/a

Table 38: MM_SET_BEACON_INT_CFM description

3.1.2.2.23 MM_SET_BASIC_RATES_REQ

This request is used to set the basic rates to the MAC HW, according to the basic rates of the current BSS.

msg_id	0x0016
Confirmation message	MM_SET_BASIC_RATES_CFM

Table 39: MM_SET_BASIC_RATES_REQ description

3.1.2.2.23.1 Parameters

Name	Type	Description
rates	u32	Basic rate set to be put into ratesReg MAC HW register. See [3] for details about this register.
inst_nbr	u8	Index of the interface for which the parameter is configured
band	u8	Band on which this interface is operating 0: 2.4GHz 1: 5GHz

Table 40: MM_SET_BASIC_RATES_REQ parameters

3.1.2.2.24 MM_SET_BASIC_RATES_CFM

This message confirms the completion of the handling of MM_SET_BASIC_RATES_REQ message.

msg_id	0x0017
Confirmation message	n/a

Table 41: MM_SET_BASIC_RATES_CFM description

3.1.2.2.25 MM_SET_BSSID_REQ

This request is used to set the BSSID to the MAC HW, according to the BSSID of the current BSS.

msg_id	0x0018
Confirmation message	MM_SET_BSSID_CFM

Table 42: MM_SET_BSSID_REQ description

3.1.2.2.25.1 Parameters

Name	Type	Description
bssid	u8[6]	BSSID of the BSS the system belongs to.
inst_nbr	u8	Index of the interface for which the parameter is configured

Table 43: MM_SET_BSSID_REQ parameters

3.1.2.2.26 MM_SET_BSSID_CFM

This message confirms the completion of the handling of MM_SET_BSSID_REQ message.

msg_id	0x0019
Confirmation message	n/a

Table 44: MM_SET_BSSID_CFM description

3.1.2.2.27 MM_SET_EDCA_REQ

This request is used to set the EDCA parameters (i.e. QoS TX parameters) to the MAC HW, according to the configuration of the current BSS.

msg_id	0x001A
Confirmation message	MM_SET_EDCA_CFM

Table 45: MM_SET_EDCA_REQ description

3.1.2.2.27.1 Parameters

Name	Type	Description
ac_param	u32	EDCA parameters for the queue being configured, to be put into edcaACxReg MAC HW register. See [3] for details about the format of this register.
uapsd	u8	Flag indicating if UAPSD is enabled for this queue 0: UAPSD disabled 1: UAPSD enabled
hw_queue	u8	Index of the HW queue being configured. 0: AC_BK. 1: AC_BE. 2: AC_VI. 3: AC_VO
inst_nbr	u8	Index of the interface for which the parameter is configured

Table 46: MM_SET_EDCA_REQ parameters

3.1.2.2.28 MM_SET_EDCA_CFM

This message confirms the completion of the handling of MM_SET_EDCA_REQ message.

msg_id	0x001B
Confirmation message	n/a

Table 47: MM_SET_EDCA_CFM description

3.1.2.2.29 MM_SET_MODE_REQ

This request is used to set the 11abgnac mode to the MAC HW, according to the current system configuration.

msg_id	0x001C
Confirmation message	MM_SET_MODE_CFM

Table 48: MM_SET_MODE_REQ description

3.1.2.2.29.1 Parameters

Name	Type	Description
abgnmode	u8	0x00: 802.11b 0x01: 802.11a 0x02: 802.11g 0x03: 802.11n-2.4GHz 0x04: 802.11n-5GHz 0x05: 802.11ac-2.4GHz 0x07: 802.11ac-5GHz

Table 49: MM_SET_MODE_REQ parameters

3.1.2.2.30 MM_SET_MODE_CFM

This message confirms the completion of the handling of MM_SET_MODE_REQ message.

msg_id	0x001D
Confirmation message	n/a

Table 50: MM_SET_MODE_CFM description

3.1.2.2.31 MM_SET_VIF_STATE_REQ

This request is used to indicate that the specified VIF is now active, i.e. that it is associated to an Access Point (in STA mode) or that the operation is started (in AP mode).

msg_id	0x001E
Confirmation message	MM_SET_VIF_STATE_CFM

Table 51: MM_SET_VIF_STATE_REQ description

3.1.2.2.31.1 Parameters

Name	Type	Description
aid	u16	AssociationId as assigned by the Access Point
active	bool	True if the VIF is now active, False otherwise
inst_nbr	u8	VIF index

Table 52: MM_SET_VIF_STATE_REQ parameters

3.1.2.2.32 MM_SET_VIF_STATE_CFM

This message confirms the completion of the handling of MM_SET_VIF_STATE_REQ message.

msg_id	0x001F
Confirmation message	n/a

Table 53: MM_SET_VIF_STATE_CFM description

3.1.2.2.33 MM_SET_SLOTTIME_REQ

This request is used to set the slot time to the MAC HW, according to the setting of the current BSS.

msg_id	0x0020
Confirmation message	MM_SET_SLOTTIME_CFM

Table 54: MM_SET_SLOTTIME_REQ description

3.1.2.2.33.1 Parameters

Name	Type	Description
slottime	u8	Slot time of the current BSS.

Table 55: MM_SET_SLOTTIME_REQ parameters

3.1.2.2.34 MM_SET_SLOTTIME_CFM

This message confirms the completion of the handling of MM_SET_SLOTTIME_REQ message.

msg_id	0x0021
Confirmation message	n/a

Table 56: MM_SET_SLOTTIME_CFM description

3.1.2.2.35 MM_SET_IDLE_REQ

This message allows controlling the IDLE state of the HW. It can be used to reduce the HW power consumption, when no operation is ongoing (e.g. no scanning, no connection), while keeping the possibility to send configuration requests.

msg_id	0x0022
Confirmation message	MM_SET_IDLE_CFM

Table 57: MM_SET_IDLE_REQ description

3.1.2.2.35.1 Parameters

Name	Type	Description
hw_idle	u8	This parameter can take the following values: 0x00: HW is active (default state when MM_START_REQ message is issued) 0x01: HW is put in IDLE mode

Table 58: MM_SET_IDLE_REQ parameters

3.1.2.2.36 MM_SET_IDLE_CFM

This message confirms the completion of the handling of MM_SET_IDLE_REQ message.

msg_id	0x0023
Confirmation message	n/a

Table 59: MM_SET_IDLE_CFM description

3.1.2.2.37 MM_KEY_ADD_REQ

This request is used to add an encryption key to the LMAC.

msg_id	0x0024
Confirmation message	MM_KEY_ADD_CFM

Table 60: MM_KEY_ADD_REQ description

3.1.2.2.37.1 Parameters

Name	Type	Description

key_idx	u8	If the key is a default key, this parameter indicates the index of this default key (from 0x00 to 0x03).
sta_idx	u8	If the key is a pairwise key, this parameter indicates the peer station index, as returned in parameters of MM_STA_ADD_CFM. If the key is a default key, this parameter has to be set to 0xFF.
key_len	u8	Length (in bytes) of the key
key	u8[32]	Key material
cipher_suite	u8	Cipher suite for the key: 0x00: WEP40 0x01: TKIP 0x02: CCMP 0x03: WEP104
inst_nbr	u8	If the key is a default key, this parameter indicates the interface index (as returned in parameters of MM_ADD_IF_CFM message) for which the key is set.
spp	u8	Indicates the type of protected A-MSDUs to be used with this key: 0x01: Payload Protected (PP) A-MSDUs 0x02: Signalling and Payload Protected (SPP) A-MSDUs 0x03: No A-MSDUs More details about how to configure this field can be found in [2], chapter 11.17.

Table 61: MM_KEY_ADD_REQ parameters

3.1.2.2.38 MM_KEY_ADD_CFM

This message confirms the completion of the handling of MM_KEY_ADD_REQ message. One of the parameters of this confirmation is the index of the key just added. This index will be used when transmitting data that has to be encrypted by the HW (set in the policy table).

msg_id	0x0025
Confirmation message	n/a

Table 62: MM_KEY_ADD_CFM description

3.1.2.2.38.1 Parameters

Name	Type	Description
status	u8	Status of the key adding procedure (0x00 if successful, different from 0 otherwise)
hw_key_idx	u8	Index of the key in the HW key memory

Table 63: MM_KEY_ADD_CFM parameters

3.1.2.2.39 MM_KEY_DEL_REQ

This request is used to remove a key from the list of encryption keys.

msg_id	0x0026
Confirmation message	MM_KEY_DEL_CFM

Table 64: MM_KEY_DEL_REQ description

3.1.2.2.39.1 Parameters

Name	Type	Description
hw_key_idx	u8	Index of the key to be removed (same index as the one returned by MM_KEY_ADD_CFM message)

Table 65: MM_KEY_DEL_REQ parameters

3.1.2.2.40 MM_KEY_DEL_CFM

This message confirms the completion of the handling of MM_KEY_DEL_REQ message.

msg_id	0x0027
Confirmation message	n/a

Table 66: MM_KEY_DEL_CFM description

3.1.2.2.41 MM_BA_ADD_REQ

This request is used to indicate to the LMAC that a BlockAck agreement has been put in place with a peer device.

msg_id	0x0028
Confirmation message	MM_BA_ADD_CFM

Table 67: MM_BA_ADD_REQ description

3.1.2.2.41.1 Parameters

Name	Type	Description

type	u8	Type of BlockAck agreement that has been put in place. Values can be 0x00 for a TX agreement, 0x01 for a RX agreement.
sta_idx	u8	Index of the STA for which a BlockAck agreement has been put in place
tid	u8	TID on which the BlockAck agreement has been put in place
bufsz	u8	Size of the reordering buffer at the peer STA
ssn	u8	Start sequence number negotiated during the BA setup

Table 68: MM_BA_ADD_REQ parameters

3.1.2.2.42 MM_BA_ADD_CFM

This message confirms the completion of the handling of MM_BA_ADD_REQ message.

msg_id	0x0029
Confirmation message	n/a

Table 69: MM_BA_ADD_CFM description

3.1.2.2.42.1 Parameters

Name	Type	Description
sta_idx	u8	Index of the STA for which a BlockAck agreement has been put in place
tid	u8	TID on which the BlockAck agreement has been put in place
status	u8	Status of the BA addition in LMAC

Table 70: MM_BA_ADD_CFM parameters

3.1.2.2.43 MM_BA_DEL_REQ

This request is used to indicate to the LMAC that a previously established TX BlockAck agreement has been deleted.

msg_id	0x002A
Confirmation message	MM_BA_DEL_CFM

Table 71: MM_BA_DEL_REQ description

3.1.2.2.43.1 Parameters

Name	Type	Description

type	u8	Type of BlockAck agreement that is deleted. Values can be 0x00 for a TX agreement, 0x01 for a RX agreement.
sta_idx	u8	Index of the STA for which a BlockAck agreement has been deleted
tid	u8	TID on which the BlockAck agreement has been deleted

Table 72: MM_BA_DEL_REQ parameters

3.1.2.2.44 MM_BA_DEL_CFM

This message confirms the completion of the handling of MM_BA_DEL_REQ message.

msg_id	0x002B
Confirmation message	n/a

Table 73: MM_BA_DEL_CFM description

3.1.2.2.44.1 Parameters

Name	Type	Description
sta_idx	u8	Index of the STA for which a BlockAck agreement has been deleted
tid	u8	TID on which the BlockAck agreement has been deleted
status	u8	Status of the BA deletion in LMAC

Table 74: MM_BA_DEL_CFM parameters

3.1.2.2.45 MM_PRIMARY_TBTT_IND

This message is forwarded to the upper MAC in AP and IBSS modes only. It indicates that the next primary TBTT is close and therefore the beacon transmission has to be programmed. If several virtual access points have been started, one beacon per VAP has to be pushed.

msg_id	0x002C
Confirmation message	n/a

Table 75: MM_PRIMARY_TBTT_IND description

3.1.2.2.46 MM_SECONDARY_TBTT_IND

This message is forwarded to the upper MAC in AP and IBSS modes only. It indicates that the next secondary TBTT is close and therefore the beacon transmission has to be programmed. If several virtual access points have been started, one beacon per VAP has to be pushed.

msg_id	0x002D

Confirmation message	n/a

Table 76: MM_PRIMARY_TBTT_IND description

3.1.2.2.47 MM_SET_POWER_REQ

This request is used to indicate to the LMAC which TX power has to be used for the transmissions self-generated by the MAC HW or the LMAC.

msg_id	0x002E
Confirmation message	MM_SET_POWER_CFM

Table 77: MM_SET_POWER_REQ description

3.1.2.2.47.1 Parameters

Name	Type	Description
inst_nbr	u8	Interface for which the parameter is set.
power	S8	Requested TX power, in dBm.

Table 78: MM_SET_POWER_REQ parameters

3.1.2.2.48 MM_SET_POWER_CFM

This message confirms the completion of the handling of MM_SET_POWER_REQ message.

msg_id	0x002F
Confirmation message	n/a

Table 79: MM_SET_POWER_CFM description

3.1.2.2.48.1 Parameters

Name	Type	Description
radio_idx	u8	Index to be set in the policy table to get the desired TX power.
power	S8	Actual TX power configured, in dBm. This value might be lower than what was requested depending on the RF capabilities.

Table 80: MM_SET_POWER_CFM parameters

3.1.2.2.49 MM_DBG_TRIGGER_REQ

This request is used to force the LMAC to trigger the embedded logic analyzer (if available), and forward a debug dump to the Upper MAC.

msg_id	0x0030
Confirmation message	n/a

Table 81: MM_SET_POWER_REQ description

3.1.2.2.50 MM_SET_PS_MODE_REQ

This request is used to configure the Power-Save state of the LMAC, MAC HW and PHY. It does not allow controlling the doze state of the HW, but it allows or disallows the LMAC putting the HW in doze state.

msg_id	0x0031
Confirmation message	MM_SET_PS_MODE_CFM

Table 82: MM_SET_PS_MODE_REQ description

3.1.2.2.50.1 Parameters

Name	Type	Description
new_state	u8	New Power-Save state requested by the Upper MAC. It can take the following values: - 0x00: Power-Save disabled - 0x01: Power-Save enabled - 0x02: Dynamic Power-Save enabled - 0x03-0xFF: Reserved

Table 83: MM_SET_PS_MODE_REQ parameters

3.1.2.2.51 MM_SET_PS_MODE_CFM

This message confirms the completion of the handling of MM_SET_PS_MODE_REQ message.

msg_id	0x0032
Confirmation message	n/a

Table 84: MM_SET_PS_MODE_CFM description

3.1.2.2.52 MM_CHAN_CTXT_ADD_REQ

This request is used to add and configure a channel context that will be link with one or several VIF.

This message is applicable only if the LMAC is compiled with the channel context option enabled.

msg_id	0x0033
Confirmation message	MM_CHAN_CTXT_ADD_CFM

Table 85: MM_CHAN_CTXT_ADD_REQ description

3.1.2.2.52.1 Parameters

Name	Type	Description
band	u8	Band (2.4GHz or 5GHz)
type	u8	Channel type: 20,40,80,160 or 80+80 MHz
prim20_freq	u16	Frequency for Primary 20MHz channel (in MHz)
center1_freq	u16	Frequency for Center of the contiguous channel or center of Primary 80+80
center2_freq	u16	Frequency for Center of the non-contiguous secondary 80+80
tx_power	s8	Maximum TX power allowed for this channel (in dBm)

Table 86: MM_CHAN_CTXT_ADD_REQ parameters

3.1.2.2.53 MM_CHAN_CTXT_ADD_CFM

This message confirms the completion of the handling of MM_CHAN_CTXT_ADD_REQ message.

This message is applicable only if the LMAC is compiled with the channel context option enabled.

msg_id	0x0034
Confirmation message	n/a

Table 87: MM_CHAN_CTXT_ADD_CFM description

3.1.2.2.53.1 Parameters

Name	Type	Description
status	u8	Status of the addition
index	u8	Index of the new channel context

Table 88: MM_CHAN_CTXT_ADD_CFM parameters

3.1.2.2.54 MM_CHAN_CTXT_DEL_REQ

This request is used to delete a channel context. It is required that all existing connections between this channel context and VIFs shall be unlinked before deleting it.

This message is applicable only if the LMAC is compiled with the channel context option enabled.

msg_id	0x0035
Confirmation message	MM_CHAN_CTXT_DEL_CFM

Table 89: MM_CHAN_CTXT_DEL_REQ description

3.1.2.2.54.1 Parameters

Name	Type	Description
index	u8	Index of the new channel context to be deleted

Table 90: MM_CHAN_CTXT_DEL_REQ parameters

3.1.2.2.55 MM_CHAN_CTXT_DEL_CFM

This message confirms the completion of the handling of MM_CHAN_CTXT_DEL_REQ message.

This message is applicable only if the LMAC is compiled with the channel context option enabled.

msg_id	0x0036
Confirmation message	n/a

Table 91: MM_CHAN_CTXT_DEL_CFM description

3.1.2.2.56 MM_CHAN_CTXT_LINK_REQ

This request is used to link a VIF with a channel context. One given channel context can be linked with several VIFs but only one channel context can be linked with a VIF. This request is rejected if the provided VIF is already linked with a channel context.

This message is applicable only if the LMAC is compiled with the channel context option enabled.

msg_id	0x0037
Confirmation message	MM_CHAN_CTXT_LINK_REQ

Table 92: MM_CHAN_CTXT_LINK_REQ description

3.1.2.2.56.1 Parameters

Name	Type	Description
vif_index	u8	VIF Index
chan_index	u8	Channel Context Index
chan_switch	u8	Flag indicating whether this is a channel switch (unlink current ctx first if true)

Table 93: MM_CHAN_CTXT_LINK_REQ parameters

3.1.2.2.57 MM_CHAN_CTXT_LINK_CFM

This message confirms the completion of the handling of MM_CHAN_CTXT_LINK_REQ message.

This message is applicable only if the LMAC is compiled with the channel context option enabled.

msg_id	0x0038

Confirmation message	n/a

Table 94: MM_CHAN_CTXT_LINK_CFM description

3.1.2.2.58 MM_CHAN_CTXT_UNLINK_REQ

This request is used to unlink a VIF and a Channel Context.

This message is applicable only if the LMAC is compiled with the channel context option enabled.

msg_id	0x0039
Confirmation message	MM_CHAN_CTXT_UNLINK_CFM

Table 95: MM_CHAN_CTXT_UNLINK_REQ description

3.1.2.2.58.1 Parameters

Name	Type	Description
vif_index	u8	VIF Index

Table 96: MM_CHAN_CTXT_UNLINK_REQ parameters

3.1.2.2.59 MM_CHAN_CTXT_UNLINK_CFM

This message confirms the completion of the handling of MM_CHAN_CTXT_UNLINK_REQ message.

This message is applicable only if the LMAC is compiled with the channel context option enabled.

msg_id	0x003A
Confirmation message	n/a

Table 97: MM_CHAN_CTXT_UNLINK_CFM description

3.1.2.2.60 MM_CHAN_CTXT_UPDATE_REQ

This request is used to update the channel parameters of provided channel context.

This message is applicable only if the LMAC is compiled with the channel context option enabled.

msg_id	0x003B
Confirmation message	MM_CHAN_CTXT_UPDATE_CFM

Table 98: MM_CHAN_CTXT_UPDATE_REQ description

3.1.2.2.60.1 Parameters

Name	Type	Description
chan_index	u8	Channel Context Index
band	u8	Band (2.4GHz or 5GHz)

type	u8	Channel type: 20,40,80,160 or 80+80 MHz
prim20_freq	u16	Frequency for Primary 20MHz channel (in MHz)
center1_freq	u16	Frequency for Center of the contiguous channel or center of Primary 80+80
center2_freq	u16	Frequency for Center of the non-contiguous secondary 80+80
tx_power	s8	Maximum TX power allowed for this channel (in dBm)

Table 99: MM_CHAN_CTXT_UPDATE_REQ parameters

3.1.2.2.61 MM_CHAN_CTXT_UPDATE_CFM

This message confirms the completion of the handling of MM_CHAN_CTXT_UPDATE_REQ message.

This message is applicable only if the LMAC is compiled with the channel context option enabled.

msg_id	0x003C
Confirmation message	n/a

Table 100: MM_CHAN_CTXT_UPDATE_CFM description

3.1.2.2.62 MM_CHAN_CTXT_SCHED_REQ

This message is applicable only if the LMAC is compiled with the channel context option enabled.

msg_id	0x003D
Confirmation message	MM_CHAN_CTXT_SCHED_CFM

Table 101: MM_CHAN_CTXT_SCHED_REQ description

3.1.2.2.62.1 Parameters

Name	Type	Description
vif_index	u8	VIF Index
chan_index	u8	Channel Context Index
type	u8	Type of the scheduling request: 0 - Normal scheduling 1 - Derogatory scheduling

Table 102: MM_CHAN_CTXT_SCHED_REQ parameters

3.1.2.2.63 MM_CHAN_CTXT_SCHED_CFM

This message confirms the completion of the handling of MM_CHAN_CTXT_SCHED_REQ message.

This message is applicable only if the LMAC is compiled with the channel context option enabled.

msg_id	0x003E
Confirmation message	n/a

Table 103: MM_CHAN_CTXT_SCHED_CFM description

3.1.2.2.64 MM_BCN_CHANGE_REQ

This message is used to modify the beacon template that is sent at each TBTT when AP mode is enabled.

This message is applicable only if the LMAC is compiled with the AUTOBCN option enabled.

msg_id	0x003F
Confirmation message	MM_BCN_CHANGE_CFM

Table 104: MM_BCN_CHANGE_REQ description

3.1.2.2.64.1 Parameters

Name	Type	Description
bcn_ptr	u32	Address of the beacon template in host memory. It will be used to download the beacon into the embedded memory
bcn_len	u16	Length of the beacon template
tim_oft	u16	Offset of the TIM IE in the beacon template (in bytes)
tim_len	u8	Length of the TIM IE in the beacon template
inst_nbr	u8	Index of the VIF for which the beacon is updated
csa_oft	u8[2]	Offset of CSA (channel switch announcement) counters (0 means no counter)

Table 105: MM_BCN_CHANGE_REQ parameters

3.1.2.2.65 MM_BCN_CHANGE_CFM

This message confirms the completion of the handling of MM_BCN_CHANGE_REQ message.

This message is applicable only if the LMAC is compiled with the AUTOBCN option enabled.

msg_id	0x0040
Confirmation message	n/a

Table 106: MM_BCN_CHANGE_CFM description

3.1.2.2.66 MM_TIM_UPDATE_REQ

This message is used to indicate to the LMAC that a PS station has traffic buffered or not on host and that the TIM bit corresponding to this station shall therefore be updated in the transmitted beacons.

This message is applicable only if the LMAC is compiled with the AUTOBCN option enabled.

msg_id	0x0041
Confirmation message	MM_TIM_UPDATE_CFM

Table 107: MM_TIM_UPDATE_REQ description

3.1.2.2.66.1 Parameters

Name	Type	Description
aid	u16	Association ID of the station for which the TIM has to be updated
tx_avail	u8	Flag indicating if some traffic is buffered on host. 0: No traffic buffered 1: Traffic buffered

Table 108: MM_TIM_UPDATE_REQ parameters

3.1.2.2.67 MM_TIM_UPDATE_CFM

This message confirms the completion of the handling of MM_TIM_UPDATE_REQ message.

This message is applicable only if the LMAC is compiled with the AUTOBCN option enabled.

msg_id	0x0042
Confirmation message	n/a

Table 109: MM_TIM_UPDATE_CFM description

3.1.2.2.68 MM_CONNECTION_LOSS_IND

This message indicates to the Upper MAC that the connection to the Access Point has been lost. This message is applicable only if the LMAC is compiled with the connection monitoring option enabled (otherwise the connection monitoring is managed by the Upper MAC).

msg_id	0x0043
Confirmation message	n/a

Table 110: MM_CONNECTION_LOSS_IND description

3.1.2.2.68.1 Parameters

Name	Type	Description
inst_nbr	u8	Index of the virtual interface that triggered the connection loss.

Table 111: MM_CONNECTION_LOSS_IND parameters

3.1.2.2.69 MM_CHANNEL_SWITCH_IND

This message indicates to the Upper MAC that the system switched to a new channel and that frame can now be transmitted on the new channel.

This message is applicable only if the LMAC is compiled with the channel context option enabled.

msg_id	0x0044
Confirmation message	n/a

Table 112: MM_CHANNEL_SWITCH_IND description

3.1.2.2.69.1 Parameters

Name	Type	Description
chan_index	u8	Index of the channel context we will switch to
roc	u8	Indicate if the switch has been triggered by a Remain on channel request
vif_index	u8	VIF on which remain on channel operation has been started (if roc == true)

Table 113: MM_CHANNEL_SWITCH_IND parameters

3.1.2.2.70 MM_CHANNEL_PRE_SWITCH_IND

This message indicates to the Upper MAC that the system is about to switch from current channel. Upon reception of this message, the Upper MAC is supposed to stop sending data on the channel.

This message is applicable only if the LMAC is compiled with the channel context option enabled.

msg_id	0x0045
Confirmation message	n/a

Table 114: MM_CHANNEL_PRE_SWITCH_IND description

3.1.2.2.70.1 Parameters

Name	Type	Description
chan_index	u8	Index of the channel context we will switch to

Table 115: MM_CHANNEL_PRE_SWITCH_IND parameters

3.1.2.2.71 MM_REMAIN_ON_CHANNEL_REQ

This request is used either to request to the LMAC to switch to provided channel and to stay on it for provided duration, or to cancel a previously start RoC procedure.

This message is applicable only if the LMAC is compiled with the channel context option enabled.

msg_id	0x0046
Confirmation message	MM_REMAIN_ON_CHANNEL_CFM

Table 116: MM_REMAIN_ON_CHANNEL_REQ description

3.1.2.2.71.1 Parameters

Name	Type	Description
op_code	u8	Operation Code: 0 - MM_ROC_OP_START: Start a Remain on Channel procedure 1 - MM_ROC_OP_CANCEL: Cancel current Remain on Channel procedure
vif_index	u8	VIF Index
band	u8	Band (2.4GHz or 5GHz)
type	u16	Channel type: 20,40,80,160 or 80+80 MHz
prim20_freq	u16	Frequency for Primary 20MHz channel (in MHz)
center1_freq	u16	Frequency for Center of the contiguous channel or center of Primary 80+80
center2_freq	u16	Frequency for Center of the non-contiguous secondary 80+80
duration_ms	u32	RoC Duration in milliseconds
tx_power	s8	Maximum TX power allowed for this channel (in dBm)

Table 117: MM_REMAIN_ON_CHANNEL_REQ parameters

3.1.2.2.72 MM_REMAIN_ON_CHANNEL_CFM

This message confirms the completion of the handling of MM_REMAIN_ON_CHANNEL_REQ message.

This message is applicable only if the LMAC is compiled with the channel context option enabled.

msg_id	0x0047
Confirmation message	n/a

Table 118: MM_REMAIN_ON_CHANNEL_CFM description

3.1.2.2.72.1 Parameters

Name	Type	Description
op_code	u8	Operation Code
status	u8	Operation status
chan_ctxt_index	u8	Channel Context Index

Table 119: MM_REMAIN_ON_CHANNEL_CFM parameters

3.1.2.2.73 MM_REMAIN_ON_CHANNEL_EXP_IND

This message indicates to the Upper MAC that the Remain on Channel duration has fully passed.

This message is applicable only if the LMAC is compiled with the channel context option enabled.

msg_id	0x0048
Confirmation message	n/a

Table 120: MM_REMAIN_ON_CHANNEL_EXP_IND description

3.1.2.2.73.1 Parameters

Name	Type	Description
vif_index	u8	VIF Index
chan_ctxt_index	u8	Channel Context Index

Table 121: MM_REMAIN_ON_CHANNEL_EXP_IND parameters

3.1.2.2.74 MM_PS_CHANGE_IND

This message indicates to the host that the given STA has indicated in its power save state.

This message is applicable only if the LMAC is compiled with the P2P GO option enabled or in FullMAC flavor.

msg_id	0x0049
Confirmation message	n/a

Table 122: MM_PS_CHANGE_IND description

3.1.2.2.74.1 Parameters

Name	Type	Description
sta_idx	u8	Index of the peer device that is switching its PS state
ps_state	u8	New PS state of the peer device 0: active 1: sleeping

Table 123: MM_PS_CHANGE_IND parameters

3.1.2.2.75 MM_TRAFFIC_REQ_IND

This message indicates to the host that the given PS STA has requested some buffered traffic (either legacy using PS-poll or UAPSD using a trigger frame).

This message is applicable only if the LMAC is compiled with the P2P GO option enabled or in FullMAC flavor.

msg_id	0x004A

Confirmation message	n/a

Table 124: MM_TRAFFIC_REQ_IND description

3.1.2.2.75.1 Parameters

Name	Type	Description
sta_idx	u8	Index of the peer device that needs traffic
pkt_cnt	u8	Number of packets that need to be sent (if 0, all buffered traffic shall be sent)
uapsd	u8	Flag indicating whether the request is for legacry PS traffic or UAPSD traffic 0: Legacy PS traffic 1: UAPSD traffic

Table 125: MM_TRAFFIC_REQ_IND parameters

3.1.2.2.76 MM_SET_PS_OPTIONS_REQ

This request is used to configure the Power-Save options of the LMAC (e.g. Listen interval).

msg_id	0x004B
Confirmation message	MM_SET_PS_OPTIONS_CFM

Table 126: MM_SET_PS_MODE_REQ description

3.1.2.2.76.1 Parameters

Name	Type	Description
vif_index	u8	Index of the VIF for which the options are set
listen_interval	u16	Listen interval to be used (0 if wake up is based on DTIM instead of listen interval)
dont_listen_bcmc	bool	Flag indicating whether the device will wait for the BC/MC traffic to be received after beacon reception or not

Table 127: MM_SET_PS_OPTIONS_REQ parameters

3.1.2.2.77 MM_SET_PS_OPTIONS_CFM

This message confirms the completion of the handling of MM_SET_PS_OPTIONS_REQ message.

msg_id	0x004C
Confirmation message	n/a

Table 128: MM_SET_PS_OPTIONS_CFM description

3.1.2.2.78 MM_P2P_VIF_PS_CHANGE_IND

This message notifies the host about an update of the P2P PS state for a given VIF. It indicates start and end of absence periods (due to Notice of Absence and Opportunistic PS features) either of the peer device if we are CLI or of ourselves if we are GO).

If new VIF PS State is 0, no more frames shall be pushed for TX on the indicated VIF.

msg_id	0x004D
Confirmation message	n/a

Table 129: MM_SET_PS_OPTIONS_CFM description

3.1.2.2.78.1 Parameters

Name	Type	Description
vif_index	u8	VIF Index
ps_state	u8	New VIF PS State

Table 130: MM_P2P_VIF_PS_CHANGE_IND parameters

3.1.2.2.79 MM_CSA_COUNTER_IND

msg_id	0x004E
Confirmation message	n/a

Table 131: MM_CSA_COUNTER_IND description

3.1.2.2.79.1 Parameters

Name	Type	Description
vif_index	u8	Index of the VIF
csa_count	u8	Updated CSA counter value

Table 132: MM_CSA_COUNTER_IND parameters

3.1.2.2.80 MM_CHANNEL_SURVEY_IND

This message is sent each time a scan period (active or passive) has ended on a given channel. It contains information about the time spent on the channel and the quality of the channel (sensed noises, idle/busy ratio).

These information can be used by host as part of an Automatic Channel Selection (ACS) mechanism in order to select the best channel to be used when starting an AP.

msg_id	0x004F
Confirmation message	n/a

Table 133: MM_CHANNEL_SURVEY_IND description

3.1.2.2.80.1 Parameters

Name	Type	Description
freq	u16	Channel Frequency
noise_dbm	i8	Noise in dBm (valid only with modem version >= 2.0)
chan_time_ms	u32	Amount of time spent on the channel in ms. 30ms if sent during an active scan, 110ms if sent during a passive scan.
chan_time_busy_ms	u32	Amount of time the primary channel was sensed busy.

Table 134: MM_CHANNEL_SURVEY_IND parameters

3.1.2.2.81 MM_BFMER_ENABLE_REQ

This message is sent after the association if the peer device claims support of VHT Beamformee (SU and optionally MU) role in its VHT capabilities.

It triggers starting of Beamformer operations in LMAC. It contains the address at which beamforming reports received from the peer device can be uploaded once received, and all missing information that shall be used during generation of NDPA (Null Data Packet Announcement), NDP (Null Data Packet) and BRP (Beamforming Report Poll) frames.

msg_id	0x0050
Confirmation message	n/a

Table 135: MM_BFMER_ENABLE_REQ description

3.1.2.2.81.1 Parameters

Name	Type	Description
host_bfr_addr	u32	Address of the report memory allocated in the host memory. It shall be used as destination address for uploads of received VHT Compressed Beamforming reports for this station.
aid	u16	Association ID (Valid only if our role is AP)
sta_idx	u8	STA Index of the connection
rx_nss	u8	Maximum number of Spatial Streams the station can receive
vht_mu_bfmee	bool	Indication if peer STA is MU Beamformer Capable

Table 136: MM_BFMER_ENABLE_REQ parameters

3.1.2.2.82 MM_SET_P2P_NOA_REQ

This message can be used in order to request to start scheduling of a Notice of Absence scheme or to stop an existing one. The provided VIF index must be the index of a P2P GO VIF.

Note that NoA started locally by LMAC due to Concurrent Mode cannot be stopped upon host request.

msg_id	0x0051
Confirmation message	MM_SET_P2P_NOA_CFM

Table 137: MM_SET_P2P_NOA_REQ description

3.1.2.2.82.1 Parameters

Name	Type	Description
vif_index	u8	Index of the VIF on which NoA has to be added/removed
noa_inst_nb	u8	Instance of the NoA to be removed, valid only if count = 0
count	u8	Number of absence to be schedule, 255 means unlimited
duration_us	u32	Absence duration in microseconds, valid only if count != 0
interval_us	u32	Absence interval in microseconds, valid only if count != 0
start_offset	u32	Offset in microseconds of first absence window from the next TBTT, valid only if count != 0

Table 138: MM_SET_P2P_NOA_REQ parameters

3.1.2.2.83 MM_SET_P2P_NOA_CFM

This message confirms the completion of the handling of MM_SET_P2P_NOA_REQ message.

This message is applicable only if the LMAC is compiled with the channel context option enabled.

msg_id	0x0052
Confirmation message	n/a

Table 139: MM_SET_P2P_NOA_CFM description

3.1.2.2.83.1 Parameters

Name	Type	Description
status	u8	Operation status

Table 140: MM_SET_P2P_NOA_CFM parameters

3.1.2.2.84 MM_SET_P2P_OPPPS_REQ

This message can be used in order to request to start or stop use of the P2P Opportunistic Power Save mode.

msg_id	0x0053
Confirmation message	MM_SET_P2P_OPPPS_CFM

Table 141: MM_SET_P2P_OPPPS_REQ description

3.1.2.2.84.1 Parameters

Name	Type	Description
u8	vif_index	Index of the VIF on which Opportunistic PS has to be started/stopped
u8	ctwindow	CT Window size in milliseconds

Table 142: MM_SET_P2P_OPPPS_REQ parameters

3.1.2.2.85 MM_SET_P2P_OPPPS_CFM

This message confirms the completion of the handling of MM_SET_P2P_OPPPS_REQ message.

This message is applicable only if the LMAC is compiled with the channel context option enabled.

msg_id	0x0054
Confirmation message	n/a

Table 143: MM_SET_P2P_OPPPS_CFM description

3.1.2.2.85.1 Parameters

Name	Type	Description
status	u8	Operation status

Table 144: MM_SET_P2P_OPPPS_CFM parameters

3.1.2.2.86 MM_P2P_NOA_UPD_IND

This message is sent to the host each time a NoA scheme is started or stopped. It allows the host to have a knowledge about the P2P PS parameters currently applied by LMAC.

msg_id	0x0055
Confirmation message	n/a

Table 145: MM_P2P_NOA_UPD_IND description

3.1.2.2.86.1 Parameters

Name	Type	Description
vif_index	u8	Index of the VIF on which NoA scheduling has been updated
noa_inst_nb	u8	Index of the NoA that has been updated

noa_type	u8	Type of NoA: - 0 = Concurrent NoA, required locally when GO interface is active in parallel with a connected STA VIF (concurrent mode) - 1 = Normal NoA, required by host
count	u8	Number of scheduled absence. If 0, the NoA has been stopped.
duration_us	u32	Absence duration in microseconds, valid only if count != 0
interval_us	u32	Absence interval in microseconds, valid only if count != 0
start_time	u32	Start time of the first absence in local time, valid only if count != 0

Table 146: MM_P2P_NOA_UPD_IND parameters

3.1.2.2.87 MM_CFG_RSSI_REQ

This message is used in order to setup a RSSI trigger to be able to indicate changes in the AP RSSI for roaming purpose.

msg_id	0x0056
Confirmation message	n/a

Table 147: MM_CFG_RSSI_REQ description

3.1.2.2.87.1 Parameters

Name	Type	Description
u8	vif_index	Index of the VIF on which the RSSI threshold has to be set
s8	rssi_thold	RSSI threshold
u8	rssi_hyst	Hysteresis to applied to the threshold to avoid multiple triggers

Table 148: MM_CFG_RSSI_REQ parameters

3.1.2.2.88 MM_RSSI_STATUS_IND

This message is asynchronously sent by the LMAC when the AP RSSI becomes higher or lower than the threshold defined using MM_CFG_RSSI_REQ. The reception of this message might trigger a roaming procedure.

msg_id	0x0057
Confirmation message	n/a

Table 149: MM_ RSSI_STATUS_IND description

3.1.2.2.88.1 Parameters

Name	Type	Description
u8	vif_index	Index of the VIF on which the RSSI has passed the threshold.
u8	rssi_status	0: RSSI has become higher than the threshold 1: RSSI has become lower than the threshold

Table 150: MM_ RSSI_STATUS_IND parameters

3.1.2.2.89 MM_CSA_FINISH_IND

This message is asynchronously sent by the LMAC to indicate that the CSA procedure is done.

msg_id	0x0058
Confirmation message	n/a

Table 151: MM_CSA_FINISH_IND description

3.1.2.2.89.1 Parameters

Name	Type	Description
u8	vif_index	Index of the VIF.
u8	status	Status of the operation
u8	chan_idx	New channel context index

Table 152: MM_CSA_FINISH_IND parameters

3.1.2.2.90 MM_CSA_TRAFFIC_IND

This message is asynchronously sent by the LMAC to indicate that the CSA procedure is in progress (resp. done) and that the traffic must be stopped (resp. restarted).

msg_id	0x0059
Confirmation message	n/a

Table 153: MM_CSA_TRAFFIC_IND description

3.1.2.2.90.1 Parameters

Name	Type	Description
u8	vif_index	Index of the VIF.
u8	enable	0: Disable the traffic 1: Enable the traffic

Table 154: MM_CSA_TRAFFIC_IND parameters

3.1.2.2.91 MM_MU_GROUP_UPDATE_REQ

This message is used to indicate to the FW that the given station is now part of some MU groups. This message has to be used only in case the FW and the HW support the Multi-User MIMO feature.

msg_id	0x005a
Confirmation message	MM_MU_GROUP_UPDATE_CFM

Table 155: MM_MU_GROUP_UPDATE_REQ description

3.1.2.2.91.1 Parameters

Name	Type	Description
u8	sta_idx	Index of the peer station that has been put in some groups
u8	group_cnt	Number of groups the station has been put into
u8, u8	groups[group_cnt]	Pairs of group_id, user_pos for each group the station now belongs to

Table 156: MM_MU_GROUP_UPDATE_REQ parameters

3.1.2.2.92 MM_MU_GROUP_UPDATE_CFM

This message confirms the completion of the handling of MM_MU_GROUP_UPDATE_REQ message.

msg_id	0x005b
Confirmation message	n/a

Table 157: MM_MU_GROUP_UPDATE_CFM description

3.1.3 Debug module API description

3.1.3.1 Overview

Table 158 gives an overview of the requests/confirmations composing the DBG API:

Request	Confirmation	Description
DBG_MEM_READ_REQ	DBG_MEM_READ_CFM	Read a location in the LMAC memory space
DBG_MEM_WRITE_REQ	DBG_MEM_WRITE_CFM	Write a location in the LMAC memory space
DBG_SET_MOD_FILTER_REQ	DBG_SET_MOD_FILTER_CFM	Set the module filter for tracing
DBG_SET_SEV_FILTER_REQ	DBG_SET_SEV_FILTER_CFM	Set the severity filter for tracing

Table 158: DBG module Control messages

3.1.3.2 Detailed description

3.1.3.2.1 DBG_MEM_READ_REQ

This request reads the memory/register location pointed by the address parameter and returns the read value.

Note that the reading is done via a 32-bit access. The sender of this message has to take care of the correct alignment of the address passed as parameter.

msg_id	0x0400
Confirmation message	DBG_MEM_READ_CFM

Table 159: DBG_MEM_READ_REQ description

3.1.3.2.1.1 Parameters

Name	Type	Description
memaddr	u32	Address of the memory/register location to be read

Table 160: DBG_MEM_READ_REQ parameters

3.1.3.2.2 DBG_MEM_READ_CFM

This message indicates the completion of the handling of DBG_MEM_READ_REQ message.

msg_id	0x0401
Confirmation message	n/a

Table 161: DBG_MEM_READ_CFM description

3.1.3.2.2.1 Parameters

Name	Type	Description
memaddr	u32	Address of the memory/register location that was read
memdata	u32	Data that was read

Table 162: DBG_MEM_READ_CFM parameters

3.1.3.2.3 DBG_MEM_WRITE_REQ

This request writes a memory/register location pointed by the address parameter.

Note that the writing is done via a 32-bit access. The sender of this message has to take care of the correct alignment of the address passed as parameter.

msg_id	0x0402
Confirmation message	DBG_MEM_WRITE_CFM

Table 163: DBG_MEM_WRITE_REQ description

3.1.3.2.3.1 Parameters

Name	Type	Description
memaddr	u32	Address of the memory/register location to be written
memdata	u32	Data to be written

Table 164: DBG_MEM_WRITE_REQ parameters

3.1.3.2.4 DBG_MEM_WRITE_CFM

This message indicates the completion of the handling of DBG_MEM_WRITE_REQ message.

msg_id	0x0403
Confirmation message	n/a

Table 165: DBG_MEM_WRITE_CFM description

3.1.3.2.4.1 Parameters

Name	Type	Description
memaddr	u32	Address of the memory/register location that was written
memdata	u32	Data that was written

Table 166: DBG_MEM_WRITE_CFM parameters

3.1.3.2.5 DBG_SET_MOD_FILTER_REQ

This request sets the per-module filter of the LMAC tracing tool. It allows the host selecting the traces of which modules it desires to get.

msg_id	0x0404
Confirmation message	DBG_SET_MOD_FILTER_CFM

Table 167: DBG_SET_MOD_FILTER_REQ description

3.1.3.2.5.1 Parameters

Name	Type	Description
mod_filter	u32	Bit field indicating the traces of which modules have to be forwarded to the upper MAC. The bits are encoded as follows: bit0: RW kernel traces bit1: DBG module traces bit2: IPC traces bit3: DMA traces bit4: MM traces bit5: TX traces bit6: RX traces bit7: PHY traces bit8-31: reserved

Table 168: DBG_SET_MOD_FILTER_REQ parameters

3.1.3.2.6 DBG_SET_MOD_FILTER_CFM

This message indicates the completion of the handling of DBG_SET_MOD_FILTER_REQ message.

msg_id	0x0405
Confirmation message	n/a

Table 169: DBG_SET_MOD_FILTER_CFM description

3.1.3.2.7 DBG_SET_SEV_FILTER_REQ

This request sets the severity filter of the LMAC tracing tool. It allows configuring which types of traces are forwarded to the host, depending on their severity.

msg_id	0x0406
Confirmation message	DBG_SET_SEV_FILTER_CFM

Table 170: DBG_SET_SEV_FILTER_REQ description

3.1.3.2.7.1 Parameters

Name	Type	Description
sev_filter	u32	Severity level of the tracing tool. This parameter can take the following values: 0x00: No traces allowed 0x01: Critical traces only 0x02: Error traces and above 0x03: Warning traces and above 0x04: Informational traces and above 0x05: All traces allowed 0x06-0xFF: reserved

Table 171: DBG_SET_SEV_FILTER_REQ parameters

3.1.3.2.8 DBG_SET_SEV_FILTER_CFM

This message indicates the completion of the handling of DBG_SET_SEV_FILTER_REQ message.

msg_id	0x0407
Confirmation message	n/a

Table 172: DBG_SET_SEV_FILTER_CFM description

3.2 Data interface

3.2.1 Transmission

3.2.1.1 Transmission flow

The transmission of MPDUs between the upper MAC and the LMAC involves the movement of 2 types of data structures:

1. The TX descriptors that include the information necessary for the LMAC to download the payloads into its buffer memory (e.g. TX buffer address in Host memory, Payload length, etc.)
2. The TX buffers that include the MPDU data and a transmission control header containing the information about how to transmit the MPDU (e.g. the rates to be used, the key used for encryption, etc.) as well as a Status field that is written by the LMAC once the transmission is complete.

The transmission flow is summarized in the figure below:

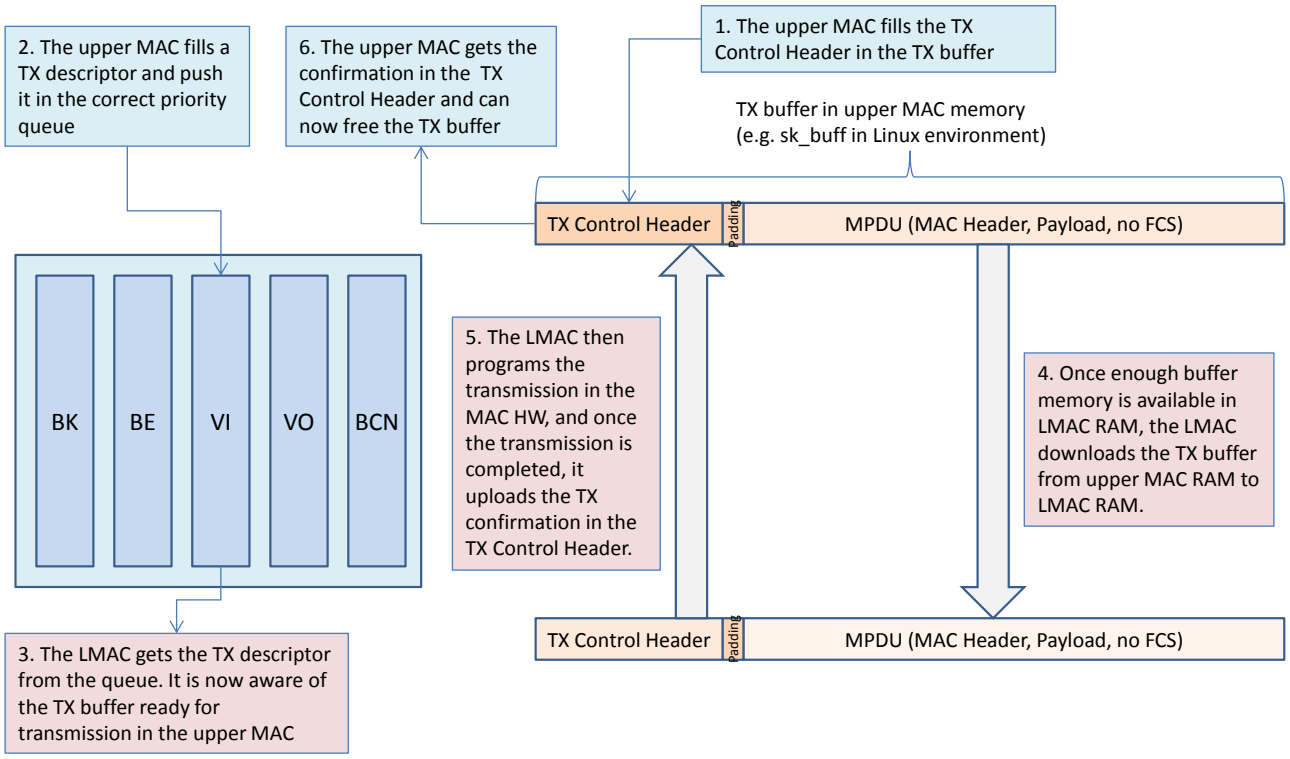


Figure 3: Transmission flow

3.2.1.2 Transmission Control Header

The TX Control Header has to be written directly in the payload buffer, just before the MAC Header. The upper MAC has therefore to reserve enough space for this header when allocating the transmission buffer.

The TX Control Header is used to control how the packet is transmitted, i.e. it contains the rate/MCS, protection, encryption key, etc. that shall be used for the transmission.

The fields of this header are the following:

Name	Type	Description
upatterntx	u32	Value has to be 0xBADCAB1E. More details about this field can be found in [3], chapter 4.3, field uPatternPT.
phycntrl1info	u32	This field has to be formatted according to [3], chapter 4.3, field PHY Control Information 1.
phycntrl2info	u32	This field has to be formatted according to [3], chapter 4.3, field PHY Control Information 2.
maccntrl1info	u32	This field has to be formatted according to [3], chapter 4.3, field MAC Control Information 1.
maccntrl2info	u32	This field has to be formatted according to [3], chapter 4.3, field MAC Control Information 2.
ratecntrl1info	u32	This field has to be formatted according to [3], chapter 4.3, field Rate Control Information 1.
ratecntrl2info	u32	This field has to be formatted according to [3], chapter 4.3, field Rate Control Information 2.
ratecntrl3info	u32	This field has to be formatted according to [3], chapter 4.3, field Rate Control Information 3.
ratecntrl4info	u32	This field has to be formatted according to [3], chapter 4.3, field Rate Control Information 4.
mac_control_info	u32	This field has to be formatted according to [3], chapter 3.1.3, field MAC Control Information 1. Bits 14 to 31 are reserved and shall be set to 0. Bits 13 shall be set to 1.
phy_control_info	u32	This field has to be formatted according to [3], chapter 3.1.3, field PHY Control Information.
status	u32	This field is written by the LMAC (using a DMA transfer) once the transmission is completed. It is formatted according to [3], chapter 3.1.3, field Status Information. It has to be initialized to 0 by the Upper MAC before being pushed to the LMAC. This field will then be used by the Upper MAC in the context of the TX confirmation checking to know if the buffer has been transmitted (if different from 0) or if it is still pending for transmission in the LMAC (if equal to 0).

Table 173: TX Control Header fields

Note: For MPDUs which are to be possibly aggregated, the **expectedACK** field in macctrlinfo1 must be set to Compressed ACK as indicated in [3].

3.2.1.3 Transmission Descriptor

The transmission descriptors are used to indicate to the LMAC the packets pending for transmission in the upper MAC memory. They have to be pushed to the LMAC after building the TX Control Header. Once a transmission descriptor has been pushed to the LMAC, the upper MAC has to wait for the transmission confirmation before freeing the TX buffer.

In order to fulfill QoS requirements, the transmission descriptors are pushed in different queues according to their priority.

The format of the transmission descriptor is described in the table below:

Name	Type	Description
packet_addr	u32	Physical address of the TX buffer in upper MAC memory. It has to point to the first byte of the TX Control Header.
packet_len	u16	Length of the MPDU. It includes the MAC Header and Payload lengths, but excludes the TX Control Header and FCS lengths.
padding	u8	Number of padding bytes between the TX Control Header and the MPDU, in the TX buffer.
tid	u8	This field is valid only for QoS frames. It contains the TID value of the QoS Control field.
staid	u8	Identifier of the station this packet is for. The identifier to be used is the one returned in the parameters of the MM_STA_ADD_CFM message. If the identifier is unknown 0xFF has to be put.
Parameters present only when Aggregation feature is enabled at compilation		
sn_win	u16	First sequence number of the BlockAck window managed by the UMAC/Driver for this staid, tid pair. This value will be put in the BAR frame sent by the MAC HW in case no BA is received after the A-MPDU transmission. Possible values from 0 to 2^{12}-1.
sn	u16	Sequence number of the MPDU, possible values from 0 to 2^{12}-1.
flags	u32	Flags from UMAC (aggregation allowed…). Use format according to [3], chapter 4.3, field MAC Control Information 2.
phy_flags	u32	Supplementary information about rate, GI, Bandwidth set for the packet. Required for calculating some AMPDU parameters BEFORE MPDU payloads are downloaded. The content of this field shall be equal to the *ratecntrl1info* field of the TX Control Header.

Table 174: TX Descriptor fields

Note: The padding information included in the TX descriptor field is used by the UMAC/Driver to indicate to the LMAC how many bytes of padding are present between the TX Control Header and the MPDU. Indeed the MPDU may not be always aligned on 32-bit words. The padding therefore allows the UMAC/Driver aligning the TX Control Header to such a boundary, in order to save some UMAC CPU cycles by doing only 32-bit accesses when writing the TX Control Header.

3.2.2 Reception

3.2.2.1 Reception flow

The reception of MPDUs involves the movement of 2 types of data structures between the LMAC and the upper MAC:

1. The upper MAC provides a list of available RX buffers in its memory.
2. Each time a new packet is received the LMAC gets a pointer to an RX buffer provided by the upper MAC, and performs a DMA transfer of the received frame to this RX buffer.

The reception flow is summarized in the figure below:

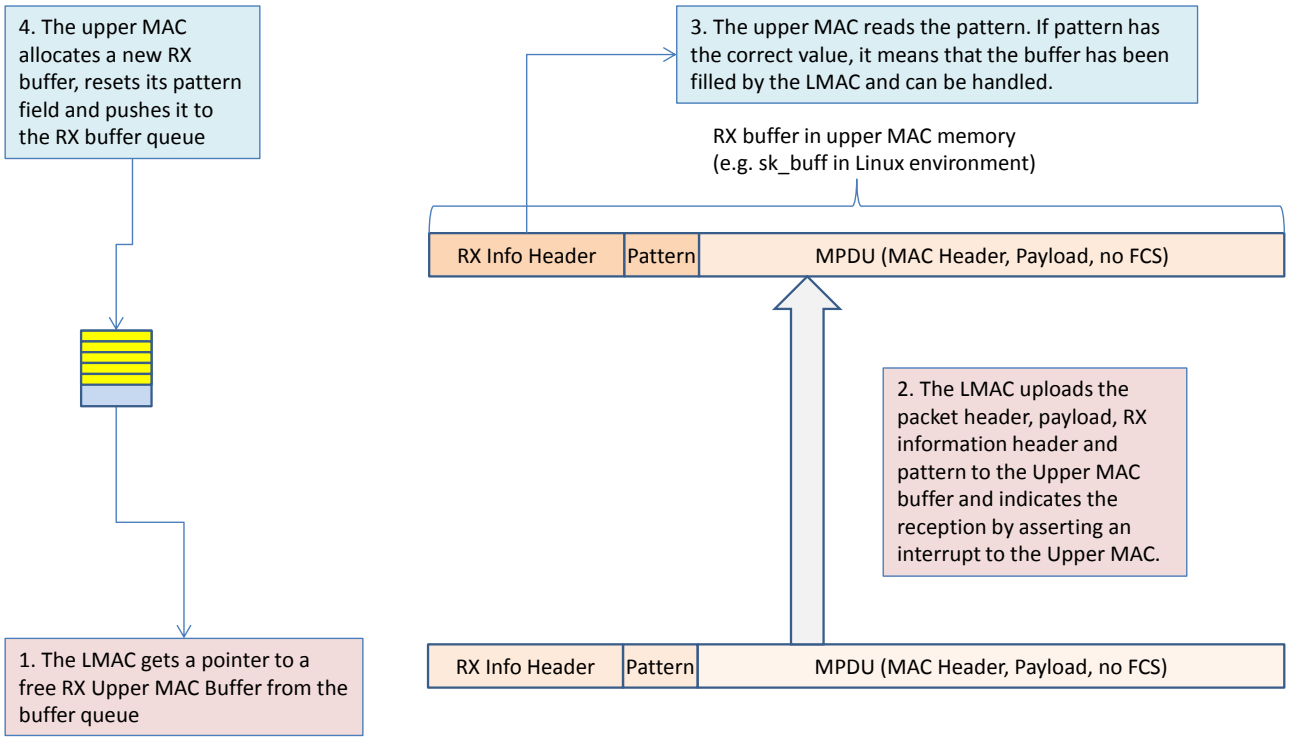


Figure 4: Reception flow

3.2.2.2 Receive Information Header

The Receive Information Header includes all the properties of the received packet. These properties include the length of the packet, its decryption status, its PHY parameters (rate/MCS, bandwidth, guard interval, channel, RSSI, etc.).

The fields of this header are the following:

Name	Type	Description
length	u32	This field contains the length of the received MPDU, including the different header length (MAC Header, IV/EIV). In case of TKIP frame reception, it also includes the TKIP MIC length. FCS, ICV or CCMP MIC lengths are not included in this field.
tsflo	u32	This field contains the lower 4 bytes of the timestamp (TSF) at which the frame ended on air.

tsfhi	u32	This field contains the higher 4 bytes of the timestamp (TSF) at which the frame ended on air.
recvec1a	u32	This field is formatted according to [3], chapter 3.2.3, field Receive Vector 1a.
recvec1b	u32	This field is formatted according to [3], chapter 3.2.3, field Receive Vector 1b.
recvec1c	u32	This field is formatted according to [3], chapter 3.2.3, field Receive Vector 1c.
recvec1d	u32	This field is formatted according to [3], chapter 3.2.3, field Receive Vector 1d.
recvec2a	u32	This field is formatted according to [3], chapter 3.2.3, field Receive Vector 2a.
recvec2b	u32	This field is formatted according to [3], chapter 3.2.3, field Receive Vector 2b.
status	u32	This field is formatted according to [3], chapter 3.2.3, field MPDU Status Information.
phychannelinfo1	u32	This field contains some information about the channel on which the frame was received. It is composed of the following elements: - Bits 31-16: prim20_freq - Bits 15-8: channel_type - Bits 7-0: band The meaning of the elements above can be found in Table 31: MM_SET_CHANNEL_REQ parameters
phychannelinfo2	u32	This field contains some information about the channel on which the frame was received. It is composed of the following elements: - Bits 31-16: center2_freq - Bits 15-0: center1_freq The meaning of the elements above can be found in Table 31: MM_SET_CHANNEL_REQ parameters

Table 175: RX Information Header fields

4 API usage

4.1 Overview

This section describes the interaction between the UMAC (or the WLAN driver in case mac80211 is the upper MAC) and the LMAC for the classical WLAN procedures. The procedures described in the following chapters are supposed to be performed starting from a system in down state (i.e. just after initialization or after applying the reset procedure as described in 4.2.1), unless stated differently.

The message sequence charts below are indicative and allow getting a good understanding of the LMAC procedures. They are not intended to provide an exhaustive list of all the allowed message combinations.

The conventions used in the message sequence charts are described in Figure 5.

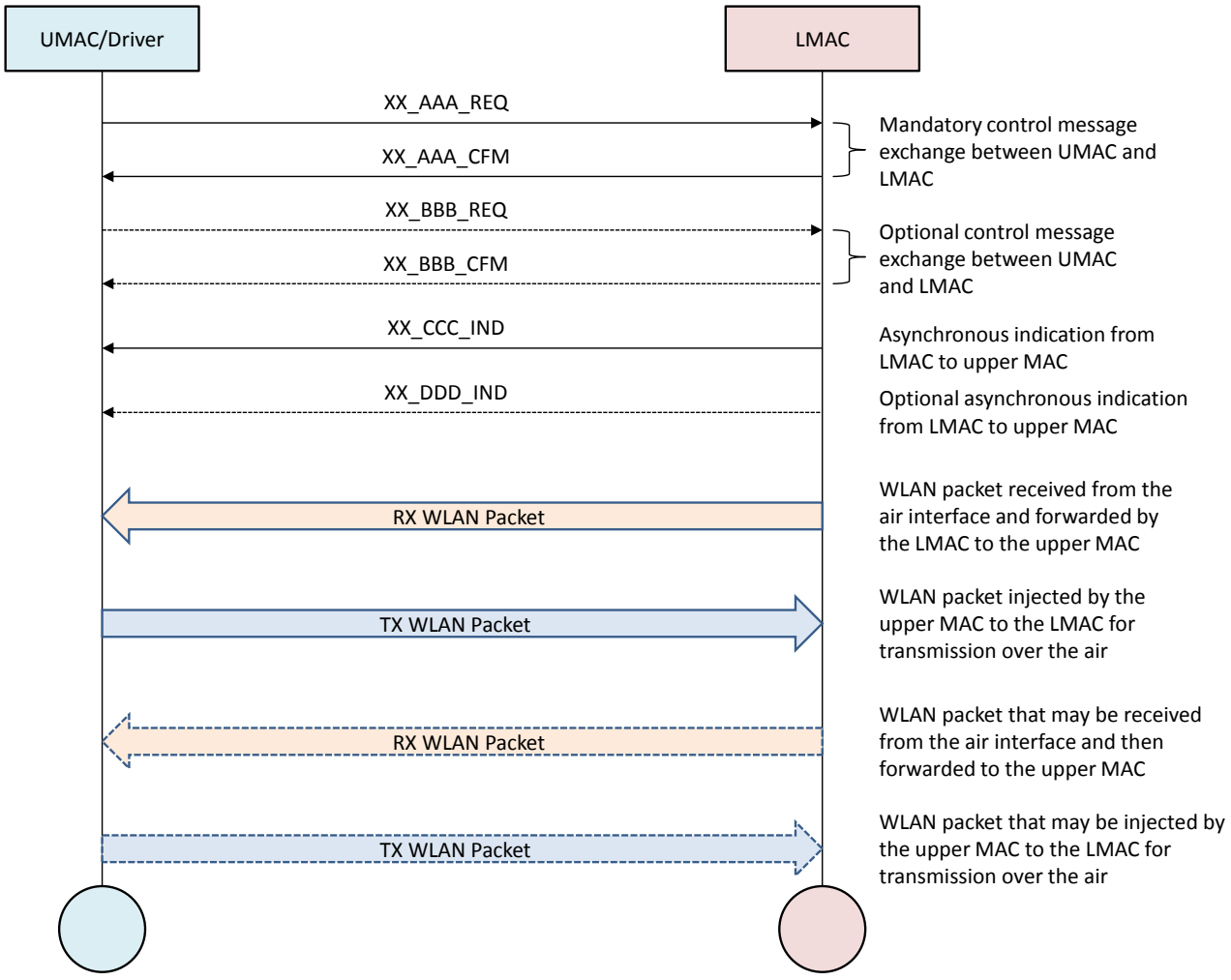


Figure 5: Message sequence chart conventions

4.2 Procedures

4.2.1 Putting the system in down state

This procedure allows putting the system in down state. It can be executed from any state, and should be executed when the WLAN interface is put in down state by the application running on top of the WLAN MAC (e.g. Linux command line "*ifconfig wlan0 down*").

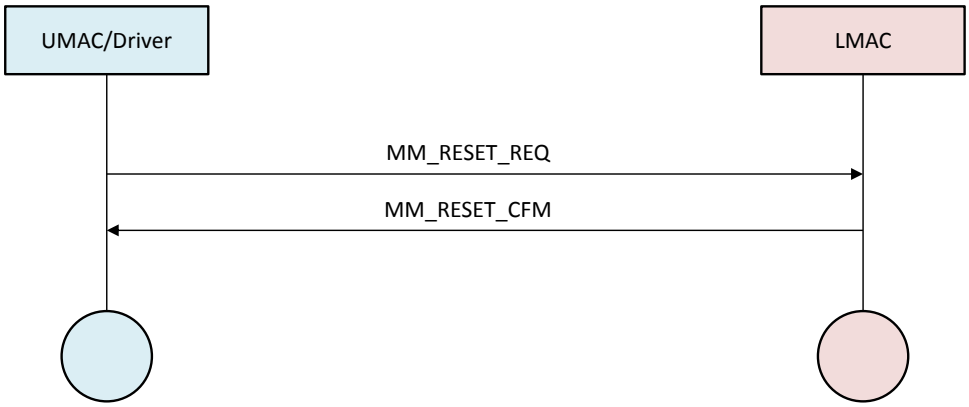


Figure 6: Putting the system in down state

4.2.2 Putting the system in monitor mode

This procedure is used to put the system in monitor mode, i.e. it allows using it as a WLAN packet sniffer. In this mode, the system has no MAC address, and will therefore not acknowledge any received packet. Packet injection is also possible, i.e. any WLAN packet injected in the LMAC is transmitted over the air.

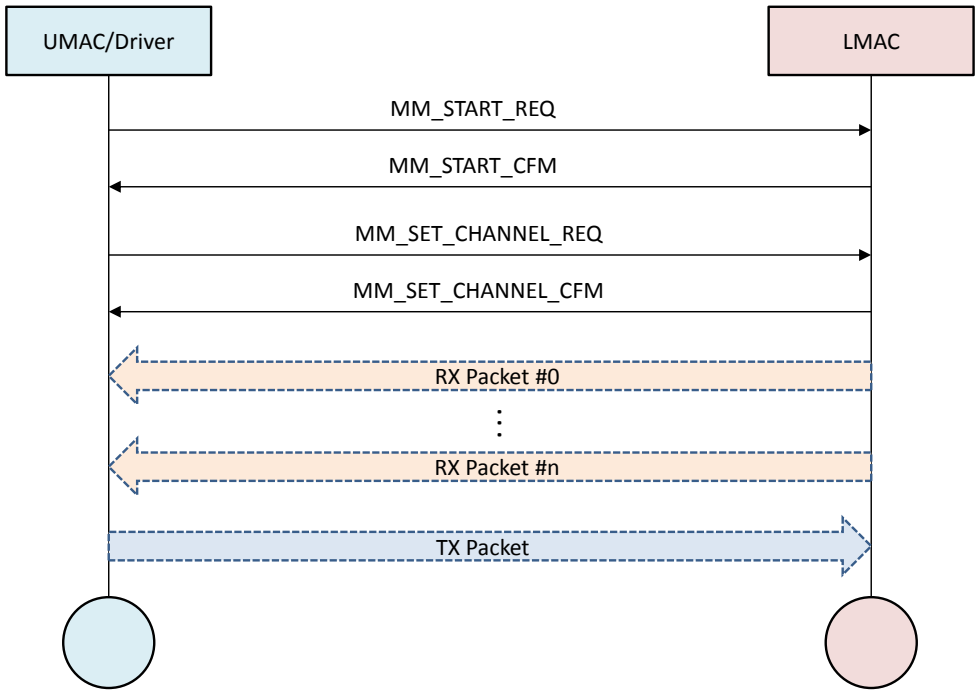


Figure 7: Putting the system in monitor mode

4.2.3 Managing the wireless interfaces

All operational modes, except the monitor mode, require the addition of a wireless interface. This action will set a MAC address to the HW, allowing it to send the acknowledgments. It also configures the HW in the correct operational mode, i.e. STA, AP or IBSS.

Once an interface has been added, the procedures listed in the following chapters can be performed.

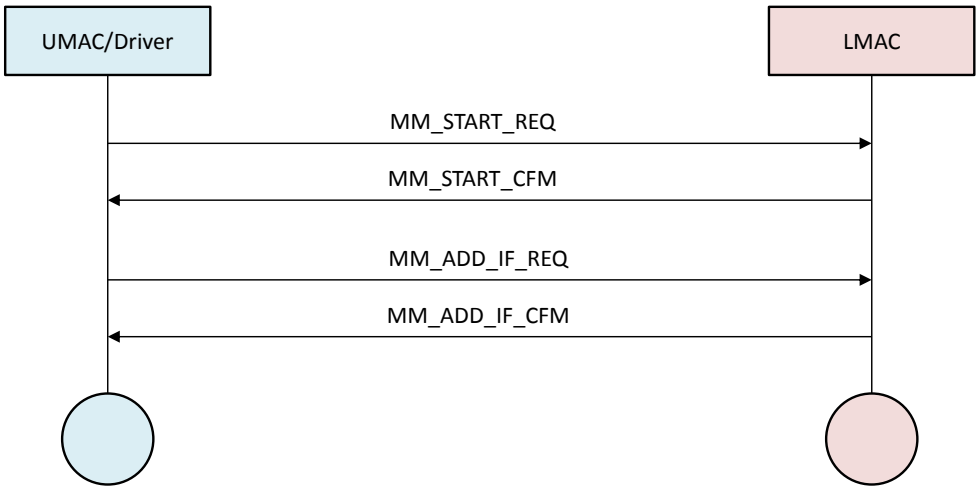


Figure 8: Adding a wireless interface

When an interface is not useful anymore, it is possible to remove it by applying the procedure below. Once all the interfaces have been removed, the system is automatically put back in monitor mode.

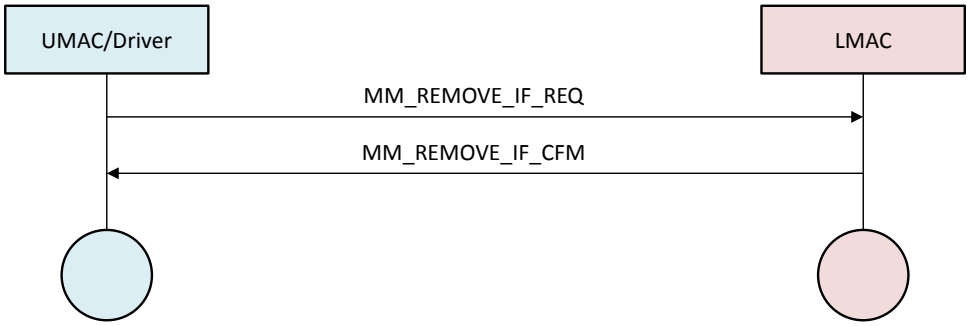


Figure 9: Removing a wireless interface

4.2.4 Managing the IDLE state of the system

The UMAC can control the IDLE state of the system. In IDLE state, the entire LMAC/MAC HW configuration is maintained, but the TX and RX paths are disabled.

When no scanning and no connection are active, this procedure can therefore be used to reduce the power consumption of the system.

The system can be moved to/out IDLE state once a MM_START_REQ has been executed.

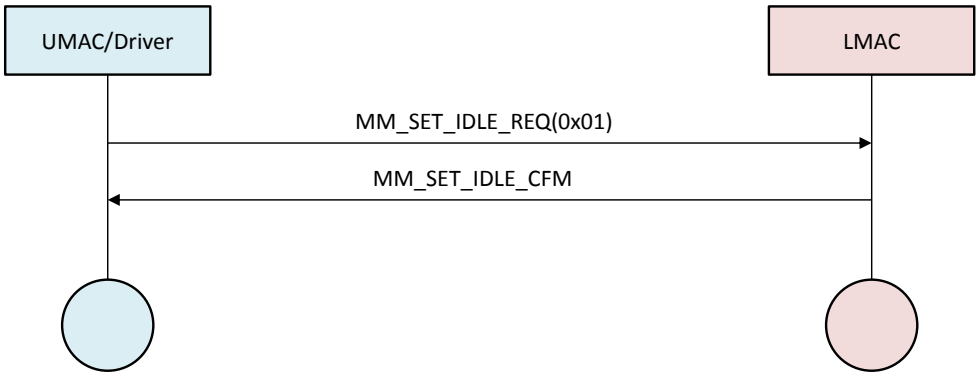


Figure 10: Moving the system in IDLE state

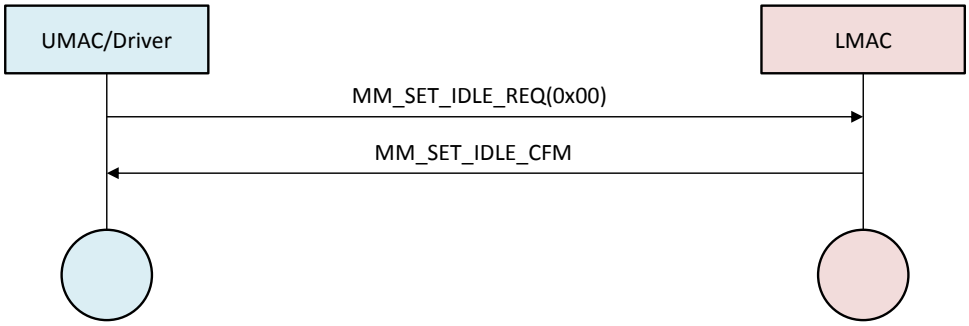


Figure 11: Moving the system out the IDLE state

4.2.5 Scanning the wireless networks

In order to discover the WLAN networks in the neighborhood, the scanning procedure has to be performed. The scanning procedure is fully managed by the upper MAC, and therefore only a few interactions with the LMAC are required.

It can be executed when at least one interface has been added to the LMAC (see 4.2.3). The sequence chart below shows the procedure including the optional transmission of a Probe Request, in case active scanning is used. The timings of the procedure (i.e. time spent on each channel) are managed by the upper MAC.

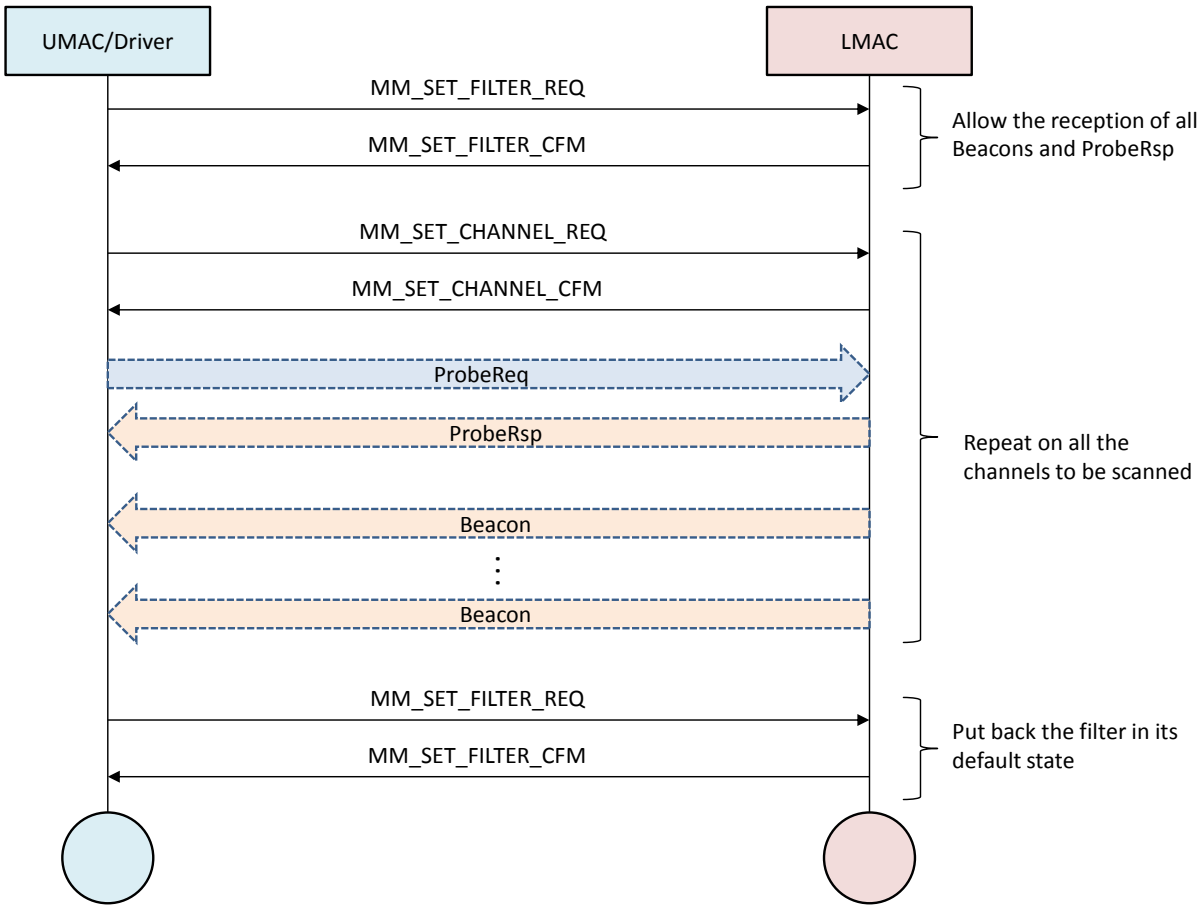


Figure 12: Scanning procedure

4.2.6 Configuring the BSS parameters

Before attempting to create or to associate to a BSS, the BSS parameters have to be configured to the LMAC/MAC HW. The sequence chart below shows the different messages used for the configuration.

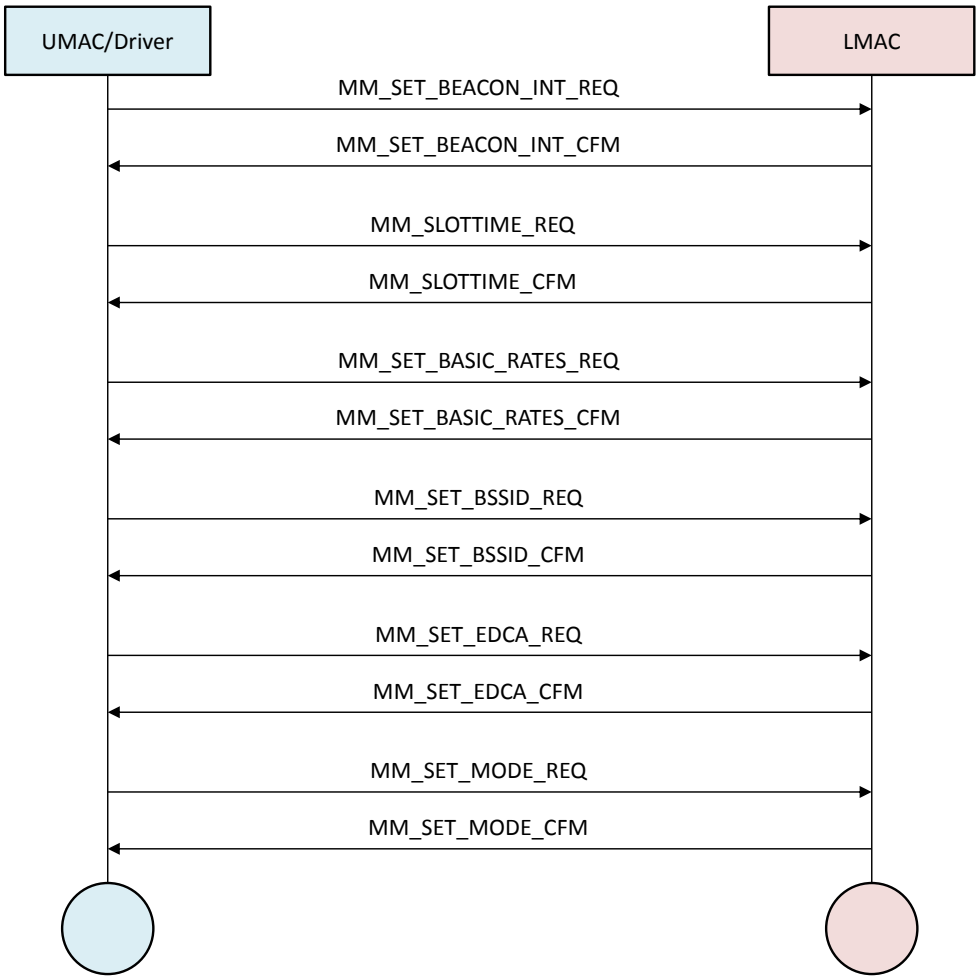


Figure 13: Configuring the BSS parameters

Some of the above message exchanges (e.g. for slot time configuration, or changes in the EDCA parameters) may be again performed when in operational mode depending on association/disassociation of new stations (AP mode) or changes in the received beacon (STA mode).

4.2.7 Managing the keys

When creating or associating to a BSS that is using protection, security keys have to be set to the MAC HW to allow the encryption and decryption of the transmitted and received packets.

Depending on the type of BSS security (WEP, WPA, WPA2), the type of key (group/default or pairwise), and the role of the device (AP or STA), the key setting procedure has to be performed at a different step:

- If WEP security is used, the WEP keys (which are default keys) are set at the time of the BSS parameter configuration (see 4.2.6)
- If WPA or WPA2 is used, two cases apply depending on the key type:
    - Pairwise Key: The key is set after the 4-way handshake, following the association procedure (see 4.2.10.1 for STA and for AP)
    - Group Key: In STA mode, the group key is set after the group key handshake, following the association procedure. In AP mode, the group key is set at the time of the BSS parameter configuration (see 4.2.6)

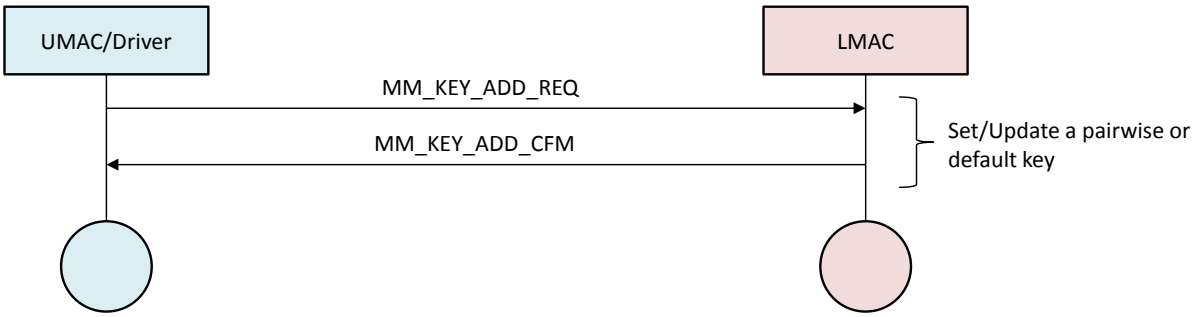


Figure 14: Key setting/updation

If required, pairwise or group keys can be removed from the MAC HW:

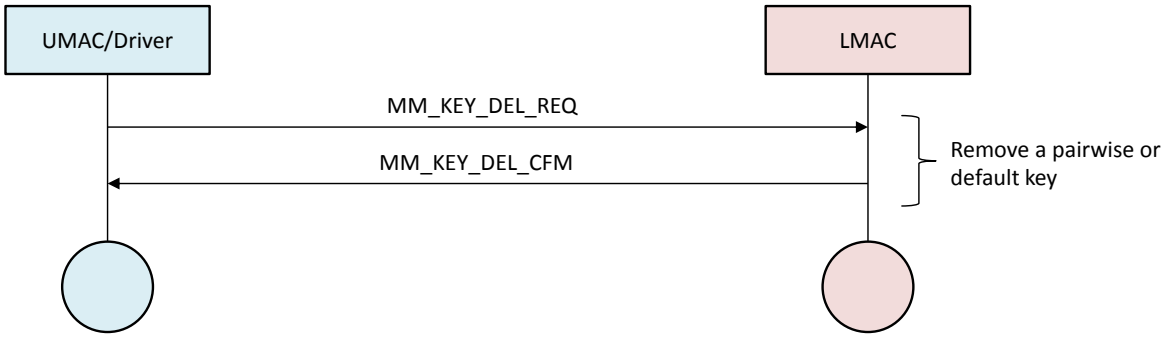


Figure 15: Key deletion

4.2.8 Managing BA agreements

The Higher Layers handle the initiation of BA agreements and its action frames. Once a BA agreement is created/destroyed, its parameters are given to LMAC for it to manage AMPDU formation.

When the pending MPDU descriptors are analysed to see if

- The Higher Layers have marked them as possible aggregate participants
- Several of them are consecutively lined up for the same (STAID, TID)
- A BA agreement exists for this (STAID, TID)
- Their sequence numbers fit into the bitmap size considered at the receiver
- Their subframes + blank delimiters byte length over the air respects the maximum AMPDU size the receiver can handle and the overall maximum AMPDU size allowed

All the information required for these checks is found in the STA Info table for the BA agreements and in the Transmission Descriptor received from UMAC/Driver.

Deletion of a BA agreement is simply done by invalidating a BA agreement parameter for that (STAID, TID). There is no allocation of space related to this operation, a small table of structures per TID for each STAID holds the BA agreement details:

- bufsz : Maximum number of AMPDUs that the receiver supports in an AMPDU (gives bitmap size)
- ssn: Starting Sequence Number – only information, the UMAC/Driver may send the first packet to be aggregated with a higher value
- sn_min: Current minimum SN which is the start of the available bitmap.

The idea while creating AMPDUs is to move sn_min in this BA agreement record as Block Ack frames are received with the bit set for Sequence Numbers ensuring no retries are still needed for smaller SNs.

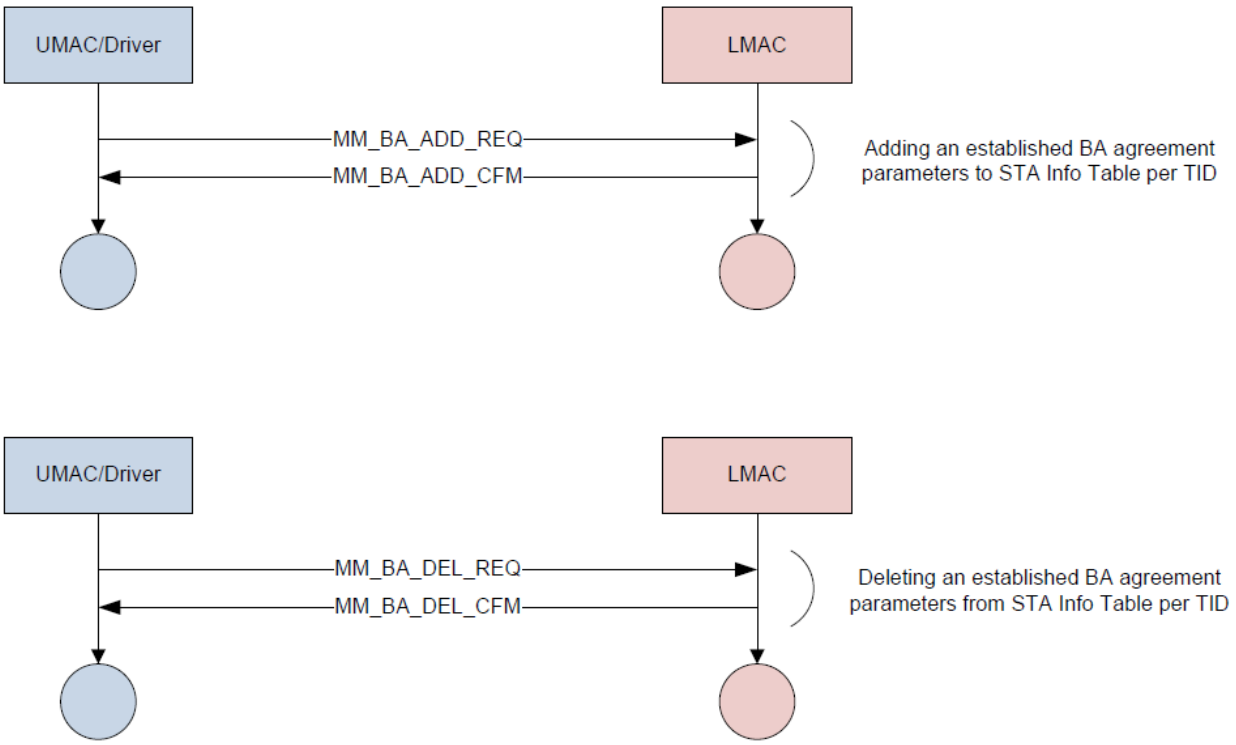


Figure 16: BA Agreement Management

4.2.9 Managing Channel Contexts

4.2.9.1 Add/Remove Channel Contexts

Before any frame transmission, a channel context has to be added and then linked with a VIF. Bandwidth is automatically allocated for a channel that is linked with at least one VIF.

Note than several VIFs can be linked with a given channel context, but only one channel context can be used by a VIF.

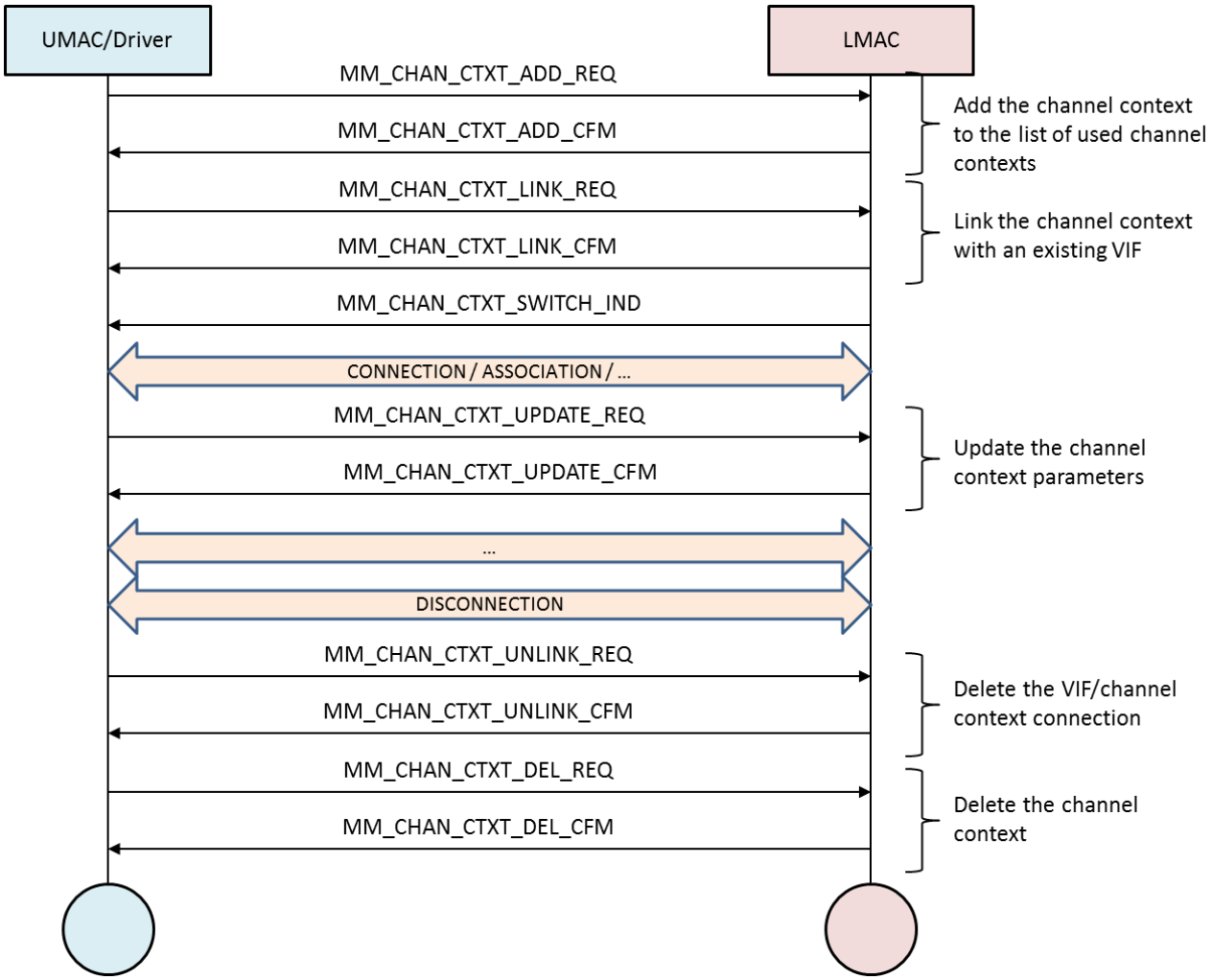
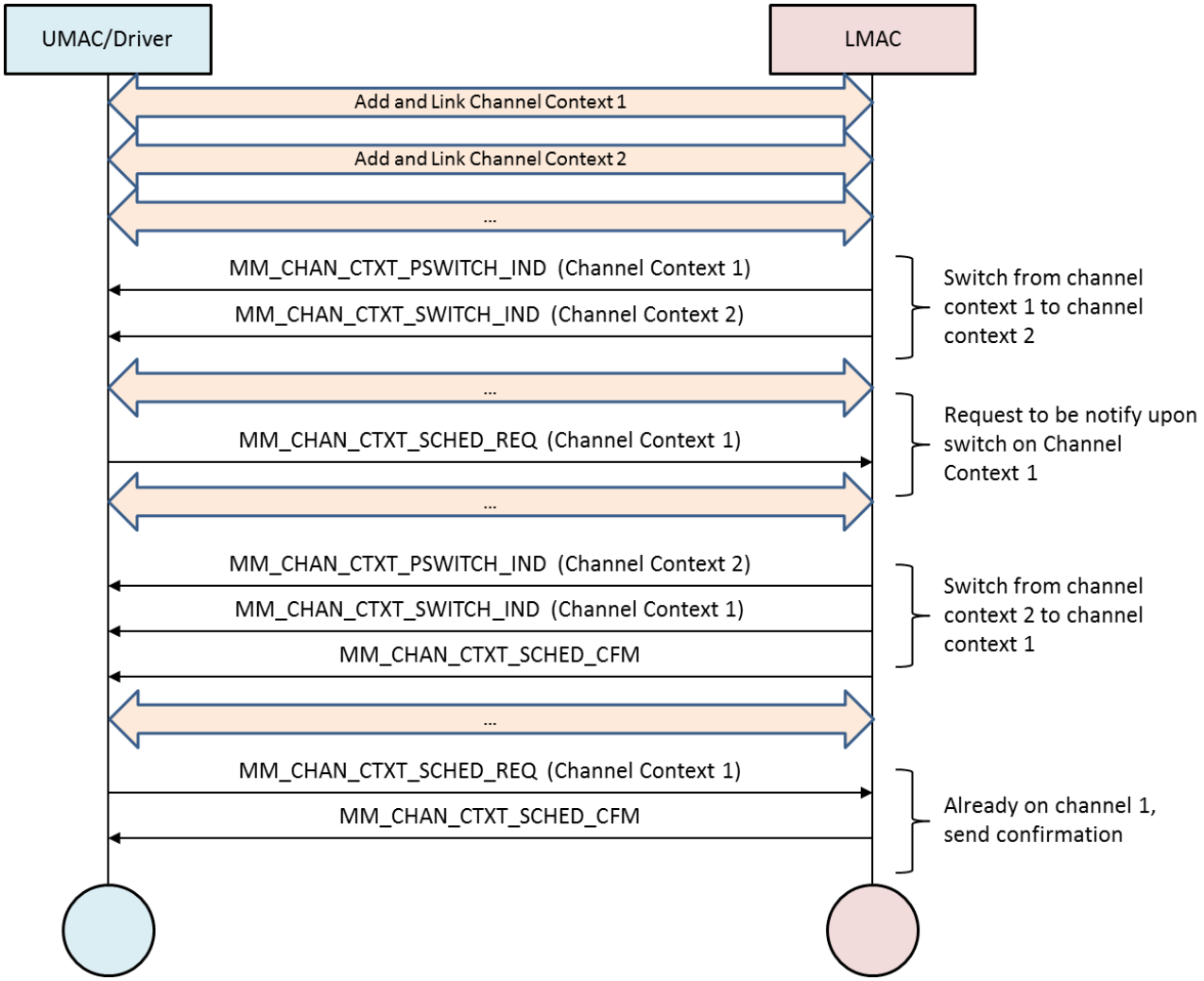


Figure 17 – Add/Remove Channel Contexts

4.2.9.2 Schedule and Switch Channel Contexts


UMAC/Driver
LMAC
Add and Link Channel Context 1
Add and Link Channel Context 2
...
MM_CHAN_CTXT_PSWITCH_IND (Channel Context 1)
MM_CHAN_CTXT_SWITCH_IND (Channel Context 2)
Switch from channel context 1 to channel context 2
...
MM_CHAN_CTXT_SCHED_REQ (Channel Context 1)
Request to be notify upon switch on Channel Context 1
...
MM_CHAN_CTXT_PSWITCH_IND (Channel Context 2)
MM_CHAN_CTXT_SWITCH_IND (Channel Context 1)
MM_CHAN_CTXT_SCHED_CFM
Switch from channel context 2 to channel context 1
...
MM_CHAN_CTXT_SCHED_REQ (Channel Context 1)
MM_CHAN_CTXT_SCHED_CFM
Already on channel 1, send confirmation

4.2.10 STA Mode

4.2.10.1 Associating to an Access Point

Once the parameters configuration procedure (see 4.2.6) has been performed, the device can associate with the desired Access Point by following the procedure below.

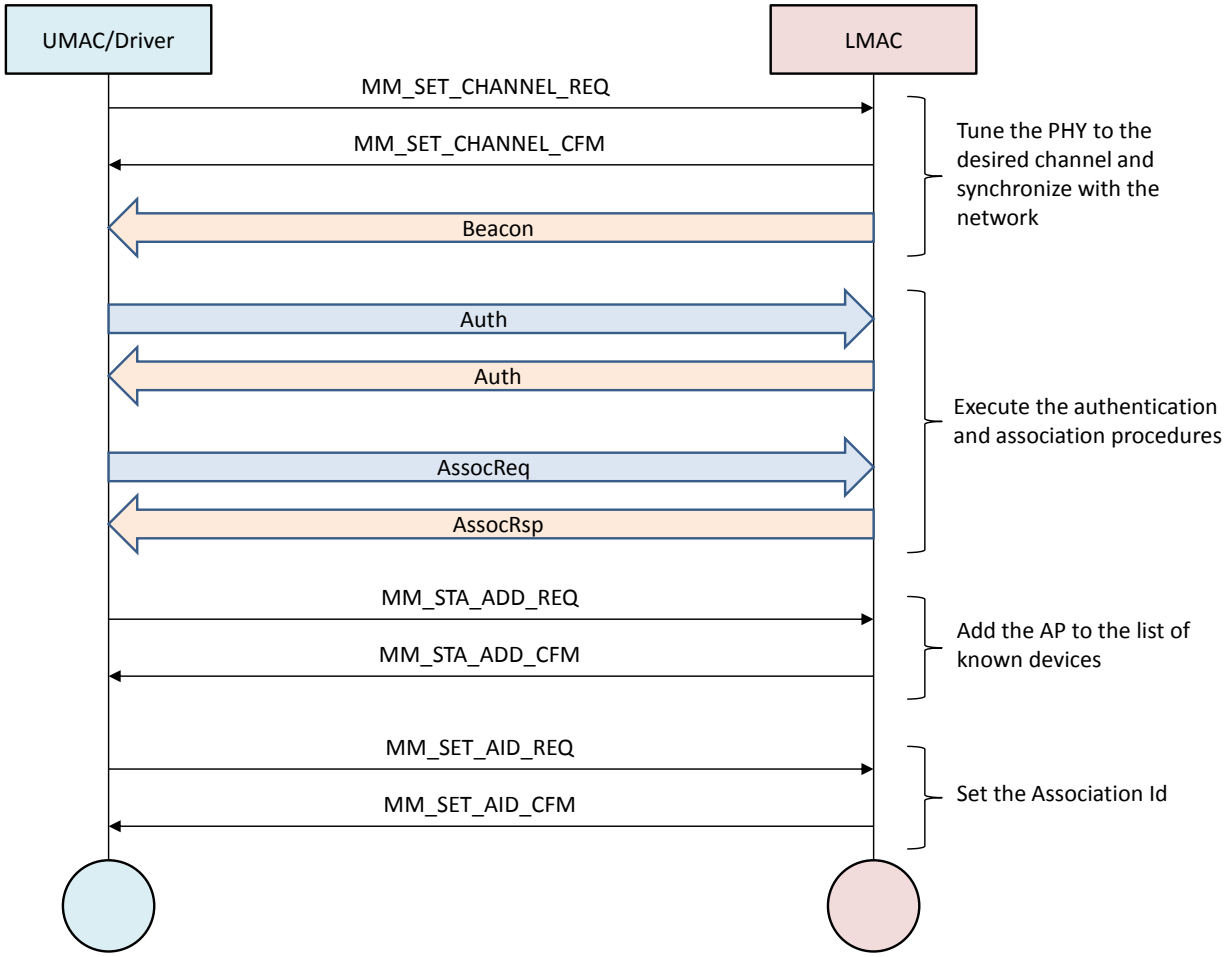


Figure 18: Association procedure

4.2.11 AP Mode

4.2.11.1 Sending the beacons

When one or more Access Point wireless interfaces are created (see 4.2.3), the system is put in Access Point mode and shall therefore start to send beacons periodically. The beacons are formatted by the UMAC and have to be transmitted at each primary and optionally, each secondary TBTT.

The LMAC/MAC HW is warning the UMAC of the need to push the beacon(s) for the next TBTT using an indication message. The UMAC then pushes the beacons for all the virtual AP interfaces created previously in the beacon transmission queue.

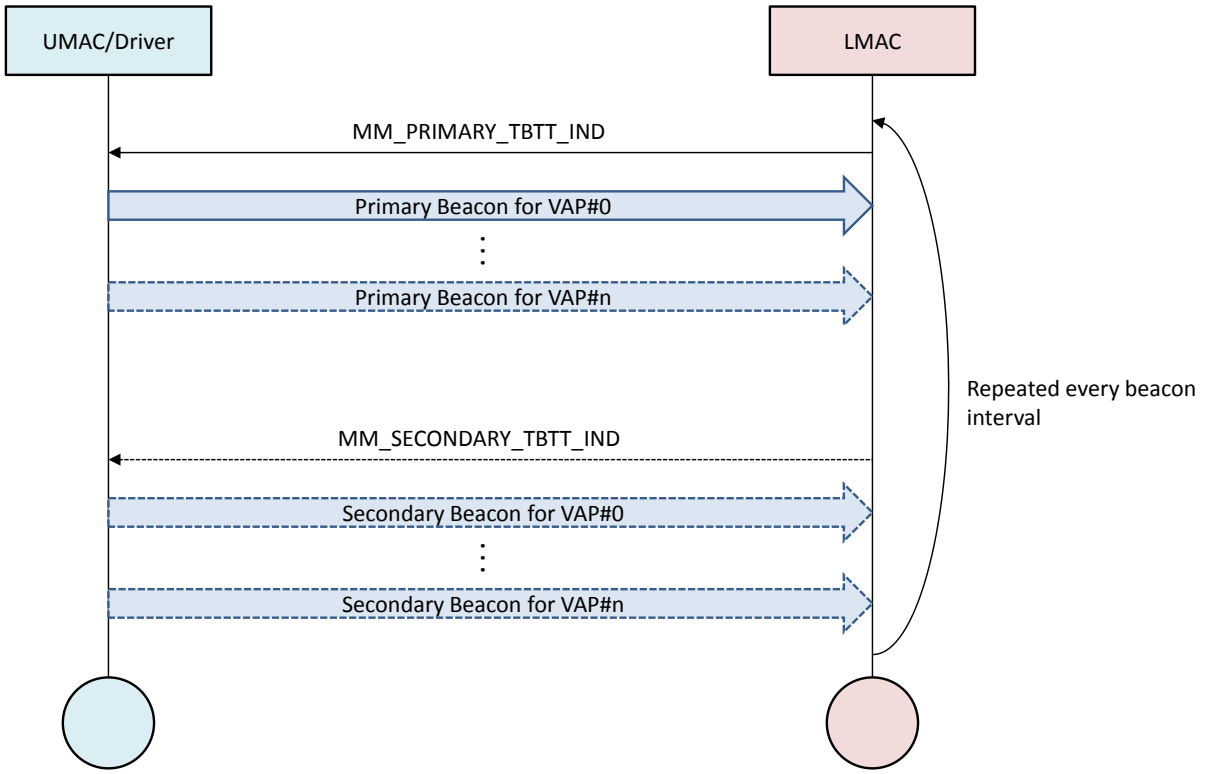


Figure 19: Beacon transmission in AP mode

4.2.11.2 Associating a station

When a STA desires to associate to the AP, the following procedure applies:

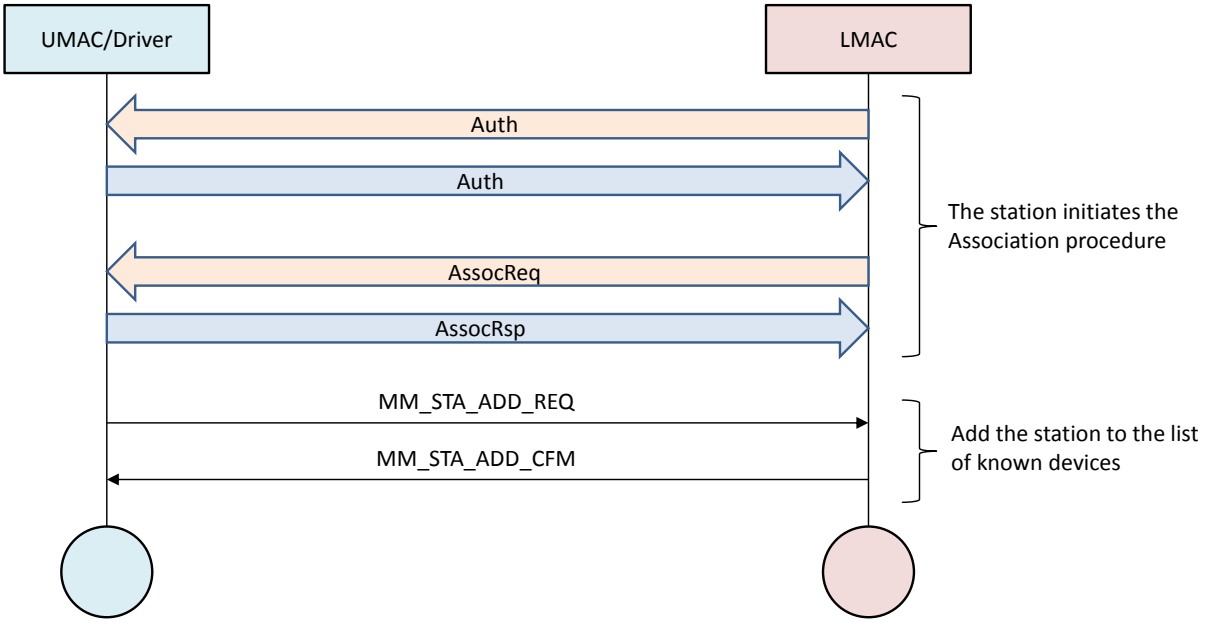


Figure 20: Associating a STA to the AP

5 Linux Control Commands

By using debugFS feature provided by the Linux kernel, the user can interact with the LMAC through the driver.

These commands may have different purposes:

- Control the LMAC behavior
- Retrieve connection information and statistics

5.1 P2P PS Modes

5.1.1 NOA Control

Use of Notice of Absence on a P2P VIF configured as GO can be controlled by entering the following command:

echo "count=$VAL1 interval=$VAL2 duration=$VAL3" > /sys/kernel/debug/ieee80211/$PHY_NAME/rwnx/p2p/noa

The table below provides a brief description of the parameters:

Name	Description
VAL1	Number of absence periods to be scheduled. 0 – Disable NoA 1-254 – Number of periods 255 – Continuous Mode
VAL2	Interval between start times of absence periods (in TUs).
VAL3	Duration of each absence period (in TUs). The following rule must be respected: (Duration <= (Interval – 10))
PHY_NAME	Name of the PHY interface on which the P2P GO VIF has been created. For example: phy0

Table 176: NOA Linux command parameters

As the LMAC only supports one P2P GO VIF, it is not required to provide the interface name.

5.1.2 OppPS Control

Use of Opportunistic Power Save mode can also be enabled or disabled with a command:

echo "ctw=VAL" > /sys/kernel/debug/ieee80211/$PHY_NAME/rwnx/p2p/oppps

The table below describes the parameters of this command:

Name	Description
VAL	CT Window duration (in TUs) If VAL=0, the OppPS will be disabled.
PHY_NAME	Name of the PHY interface on which the P2P GO VIF has been created. For example: phy0

Table 177: OppPS Linux command parameters

As the LMAC only supports one P2P GO VIF, it is not required to provide the interface name.

References

[1] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, IEEE Std 802.11™-2007

[2] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Enhancements for Higher Throughput, IEEE Std 802.11n™-2009

[3] RW-WLAN-nX-MAC-HW User Manual Document (RW-WLAN-nX-MAC-HW-UM/1.03)