1.1 Transmit path specification

Editorial note: This is clause

1.1.1 Principles of operation

The transmit path of the VLC sublayer includes the Transmit process. The Transmit process waits for assertion of the VLCSI:MA_DATA.request, VLCSI:VLCPDU.request, or VLCSI:OMCI.request primitives.

Upon assertion of the VLCSI:VLCPDU.request primitive, the received parameters are encapsulated into a VLCPDU with subtype `VLC_CONFIG` (see **Error! Reference source not found.**) according to the format defined in 8.1.1. Conceptually, this action takes place in the VLC Interface adapter as shown in Figure 6-1. Note that both the MAC destination address and the MAC source address are equal to the local MAC address assigned to the port to which the VLC sublayer is associated. The resulting VLCPDU is supplied to the egress CTE.

Upon assertion of the VLCSI:OMCI.request primitive, the received parameters are encapsulated into a VLCPDU with subtype `OMCI_SUBTYPE` (see **Error! Reference source not found.**) according to the format defined in 5.2.3. Conceptually, this action takes place in the OMCI Interface adapter as shown in Figure 6-1. Note that both the MAC destination address and the MAC source address are equal to the local MAC address assigned to the port to which the VLC sublayer is associated. The resulting VLCPDU is supplied to the egress CTE.

Upon assertion of the VLCSI:MA_DATA.request primitive, the received parameters are supplied to the egress CTE.

After the above processes are complete, the resulting xPDU is processed by the Egress Classification and Translation Engine (CTE). If a match is found, the frame is modified according to the matched rule’s action. If the frame does not match any rules, it is passed through the CTE block unmodified.

Note that to enter a tunnel, the VLC xPDU or the OMCI xPDU require a matching egress CTE rule that, at a minimum, overwrites the local MAC address value in the VLCPDU destination address field with the MAC address associated with the xPDU destination for the given tunnel.

1.1.2 Constants

The constants referenced in this state diagram are defined in **Error! Reference source not found.**.

1.1.3 Variables

`EgressRuleId`

TYPE: 16-bit unsigned integer

This variable identifies one of the provisioned CTE egress rules. It also may have a special value `none` that does not identify any of the provisioned rules.

`MaDataTxInput`

TYPE: structure

This variable contains the set of parameters of the VLCSI:MA_DATA.request() primitive as defined in 4.3.1.1.1.

`VlcCfgTxInput`

TYPE: structure

This variable contains the set of parameters of the VLCSI:VLCPDU.request() primitive as defined in 4.3.1.2.1.

`OmciTxInput`

TYPE: structure

This variable contains the set of parameters of the VLCSI:OMCI.request() primitive as defined in 4.3.1.3.1.

`TxInputPdu`

TYPE: structure

This variable holds an Ethernet frame to be passed to the CTE. The fields of this structure correspond to the parameters of the `MA_DATA.request()` primitive, as defined in IEEE Std 802.3, 2.3.1. It is formed as the result of receiving input from the VLCSI:OMCI.request(), VLCSI:VLCPDU.request(), or VLCSI:MA_DATA.requst() primitives and is passed as the input to the CTE.

The `TxInputPdu` structure supports the `AddField(field_code, field_value)` method. The `field_code` parameter takes values as defined in **Error! Reference source not found.**.

`TxOutputPdu`

TYPE: structure

This variable holds an Ethernet frame to be passed to the MACCSI:MA_DATA interface. The fields of this structure correspond to the parameters of the `MA_DATA.request()` primitive, as defined in IEEE Std 802.3, 2.3.1.

1.1.4 Functions

`CheckEgressRules(input_pdu)`

This function returns the identification of one and only one egress rule that matches the frame contained in the `input_pdu` structure. It is out of the scope of this standard to specify how this function chooses its return value if multiple rules match the frame. If none of the rules matches the frame, a special value, `none`, is returned

`Modify(rule_id, input_pdu)`

This functions is defined in **Error! Reference source not found.**.

`Concat(value1, value2, …, valueN)`

This function returns the concatenation of the input parameters. The input parameters are concatenated in the order they appear in the function call.

1.1.5 Primitives

The primitives referenced in this state diagram are defined in **Error! Reference source not found.**.

1.1.6 State Diagram

VLC sublayer shall implement the Transmit process as defined in the state diagram in Figure **Error! No text of specified style in document.**-1.

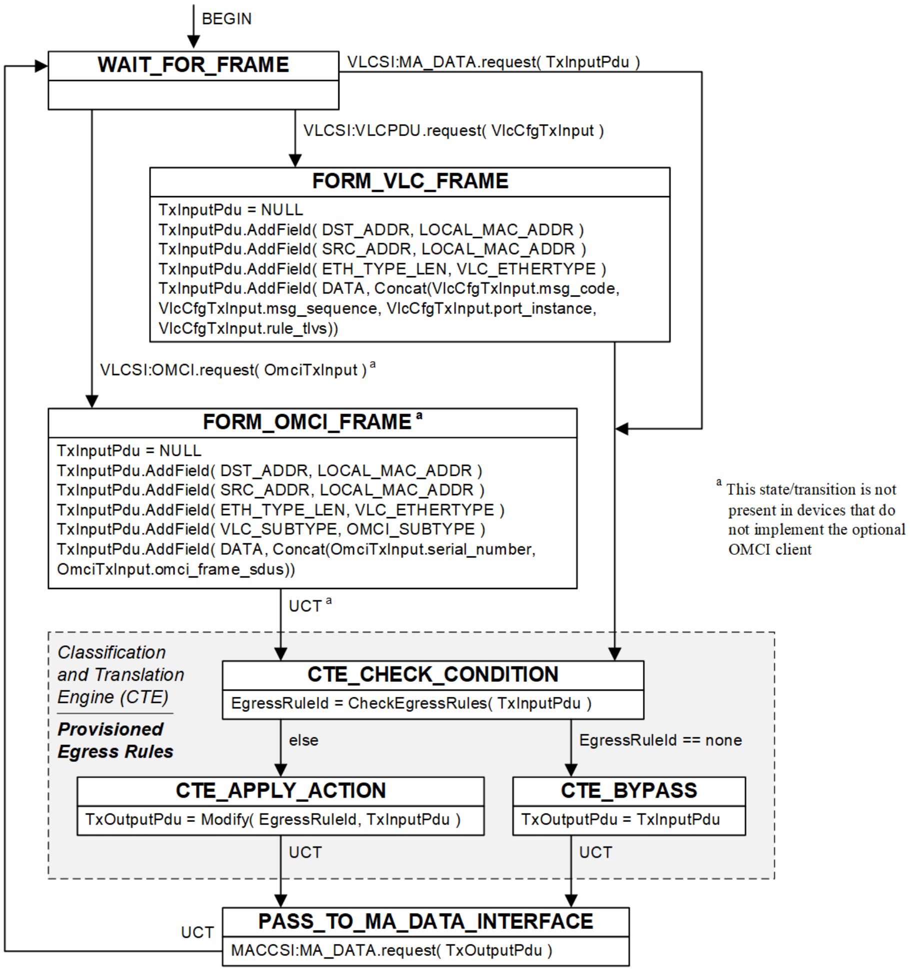
BEGIN
WAIT_FOR_FRAME
VLCSI:MA_DATA.request(TxInputPdu)
VLCSI:VLCPDU.request(VlcCfgTxInput)
FORM_VLC_FRAME
TxInputPdu = NULL
TxInputPdu.AddField(DST_ADDR, LOCAL_MAC_ADDR)
TxInputPdu.AddField(SRC_ADDR, LOCAL_MAC_ADDR)
TxInputPdu.AddField(ETH_TYPE_LEN, VLC_ETHERTYPE)
TxInputPdu.AddField(DATA, Concat(VlcCfgTxInput.msg_code, VlcCfgTxInput.msg_sequence, VlcCfgTxInput.port_instance, VlcCfgTxInput.rule_tlvs))
VLCSI:OMCI.request(OmciTxInput) a
FORM_OMCI_FRAME a
TxInputPdu = NULL
TxInputPdu.AddField(DST_ADDR, LOCAL_MAC_ADDR)
TxInputPdu.AddField(SRC_ADDR, LOCAL_MAC_ADDR)
TxInputPdu.AddField(ETH_TYPE_LEN, VLC_ETHERTYPE)
TxInputPdu.AddField(VLC_SUBTYPE, OMCI_SUBTYPE)
TxInputPdu.AddField(DATA, Concat(OmciTxInput.serial_number, OmciTxInput.omci_frame_sdus))
a This state/transition is not present in devices that do not implement the optional OMCI client
UCT a
Classification and Translation Engine (CTE)
Provisioned Egress Rules
CTE_CHECK_CONDITION
EgressRuleId = CheckEgressRules(TxInputPdu)
else
EgressRuleId == none
CTE_APPLY_ACTION
TxOutputPdu = Modify(EgressRuleId, TxInputPdu)
CTE_BYPASS
TxOutputPdu = TxInputPdu
UCT
UCT
UCT
PASS_TO_MA_DATA_INTERFACE
MACCSI:MA_DATA.request(TxOutputPdu)

Figure Error! No text of specified style in document.**-1—Transmit process state diagram**