6 VLC Sublayer

6.1 VLC Functional Block Diagram

6.2 VLC Classification and Translation Engine

6.3 Receive path specification

6.3.1 Principles of operation

The receive path of the VLC sublayer includes the Receive process. The Receive process waits for a frame to be received on *MACCSI:MA_DATA* interface (via `MACCSI:MA_DATA.indication()` primitive as defined in 4.3.1.1.2). When a frame is received, it is processed by the ingress Classification and Translation Engine (CTE) and if a match is found, the frame is modified according to the matched rule action. If the frame does not match any rules, it is passed through the CTE block unmodified.

After traversing the ingress CTE block (highlighted in Figure 6-4), the frame is dispatched to one of the VLCSI interfaces: (*VLCSI:VLCPDU*, *VLCSI:OMCI*, or *VLCSI:MA_DATA*). The dispatching decision is based on the values of the MAC destination address, Ethertype, and VLC subtype.

VLCPDUs with the destination address matching the local MAC address and the VLC subtype equal to `SUBTYPE_VLC` (see Table 5-1) are modified to match the parameters expected by the `VLCSI:VLCPDU.indication()` primitive (see 4.3.1.3.2) and are passed to the *VLCSI:VLCPDU* interface. In addition, the source address of these VLCPDUs is stored and used later as the default destination address for transmitted VLCPDUs with `SUBTYPE_VLC` subtype.

VLCPDUs with the destination address matching the local MAC address and the VLC subtype equal to `SUBTYPE_OAM` (see Table 5-1) are converted into OAMPDUs and are passed to the *VLCSI:MA_DATA* interface.

VLCPDUs with the destination address matching the local MAC address and the VLC subtype equal to `SUBTYPE_OMCI` (see Table 5-1) are modified to match the parameters expected by the `VLCSI:OMCI.indication()` primitive (see 4.3.1.4.2) and are passed to the *VLCSI:OMCI* interface. In addition, the source address of these VLCPDUs is stored and used later as the default destination address for transmitted VLCPDUs with `SUBTYPE_OMCI` subtype.

All other xPDUs are passed unmodified to the *VLCSI:MA_DATA* interface. Note that there still may be other local clients that are able to intercept/consume these xPDUs at a higher layer.

The Receive process does not discard any frames, i.e., every `MACCSI:MA_DATA.indication()` primitive results in a generation of a single indication primitive on either *VLCSI:VLCPDU*, *VLCSI:OMCI*, or *VLCSI:MA_DATA* interface.

Note that no provisioning of the ingress tunnel exit rules is required in situations where the tunnel is terminated at the same port where the xPDUs are to be consumed by their respective clients. The functionality to convert VLCPDUs into xPDUs is built-in into the Receive process.

6.3.2 Constants

ETHERTYPE_SP

This constant holds the value of the Ethertype identifying the Slow Protocol (see IEEE Std 802.3, 57A.4).

ETHERTYPE_VLC

TYPE: 16-bit Ethertype

This constant holds the Ethertype value identifying the VLCPDUs.

VALUE: 0xA8-C8

LOCAL_MAC_ADDR

TYPE: 48-bit MAC address

This constant holds the value of the MAC address associated with the port where the Receive process state diagram is instantiated. Some devices may associate the same MAC address value with multiple ports. The format of the MAC address is defined in IEEE Std 802.3, 3.2.3.

VALUE: device-specific

NULL_MAC_ADDR

TYPE: 48-bit MAC address

This constant holds the placeholder value of destination MAC address, before the actual destination address value was determined or configured for the given frame. Frames witrh the DstAddr field equal to the NULL_MAC_ADDR are not transmitted by a compliant device.

VALUE: 0x00-00-00-00-00-00

SP_ADDR

This constant holds the value of the destination MAC address associated with Slow Protocols (see IEEE Std 802.3, 57A.3).

SUBTYPE_OAM

This constant represents the value of the VLC subtype that identifies OAM payload carried within a VLCPDU as defined in Table 5-1.

SUBTYPE_OMCI

This constant represents the value of the VLC subtype that identifies OMCI payload carried within a VLCPDU as defined in Table 5-1.

SUBTYPE_VLC

This constant represents the value of the VLC subtype that identifies *VLC_CONFIG* VLCPDU as defined in Table 5-1.

6.3.3 Variables

DstAddress

This variable represents the DstAddress field as defined in Table 6-2.

LengthType

This variable represents the LengthType field as defined in Table 6-2.

IngressRuleId

TYPE: 16-bit unsigned integer

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

OmciPeerAddr

TYPE: 48-bit MAC address

The OmciPeerAddr variable holds the MAC address value of the OMCI peer entity. This variable is shared among the Receive process and the Transmit process. At initialization, this variable is assigned the default value of NULL_MAC_ADDR. When a VLCPDU with the subtype SUBTYPE_OMCI is received, this variable is set to the value of the SrcAddrs field of that VLCPDU.

RxInputPdu

TYPE: structure containing an Ethernet frame

This variable holds an Ethernet frame received from the *MACCSI:MA_DATA* interface. The fields of this structure correspond to the parameters of the MA_DATA.indication() primitive as defined in IEEE Std 802.3, 2.3.2.

RxOutputPdu

TYPE: structure containing an Ethernet frame

This variable holds an Ethernet frame to be passed to one of the the VLCSI interfaces (*VLCSI:VLCPDU*, *VLCSI:OMCI*, or *VLCSI:MA_DATA*). The fields of this structure correspond to the parameters of the MA_DATA.indication() primitive as defined in IEEE Std 802.3, 2.3.2.

Additionally, the RxOutputPdu structure supports the RemoveField(field) method, which removes the field from the structure. Thus, unlike the RxInputPdu structure, the RxOutputPdu may contain only a partial Ethernet frame. The field parameter retprepresents one of the frame fields defined in Table 6-2.

SrcAddress

This variable represents the SrcAddress field as defined in Table 6-2.

Subtype

This variable represents the Subtype field as defined in Table 6-2.

VlcPeerAddr

TYPE: 48-bit MAC address

The VlcPeerAddr variable holds the MAC address value of the VLC peer entity. This variable is shared among the Receive process and the Transmit process. At initialization, this variable is assigned the default value of NULL_MAC_ADDR. When a VLCPDU with the subtype SUBTYPE_VLC is received, this variable is set to the value of the SrcAddrs field of that VLCPDU.

6.3.4 Functions

`CheckIngressRules(input_pdu)`

This function returns the identification of an ingress rule that matched the frame contained in the `RxInputPdu` structure. If multiple rules match a frame, the function returns a single identification of any of these rules. The selection criteria is vendor-specific and outside the scope of this standard. If none of the rules matches the frame, a special value `none` is returned.

`Modify(rule_id, input_pdu)`

This function returns a frame that is a result of applying the modification action(s) of the rule identified by the `rule_id` parameter to the frame contained in the `input_pdu` parameter.

`RxOutputPdu.ReplaceField(target_field, source_field)`

This function is a method associated with `RxOutputPdu` structure used in the Receive process state diagram. This method replaces the value of a field in the structure, specified by `target_field`, with the value from the field specified by `source_field`. The `target_field` and `source_field` parameters are one of the frame fields defined in Table 6-2.

`RxOutputPdu.RemoveField(field)`

This function is a method associated with `RxOutputPdu` structure used in the Receive process state diagram. This method removes the `field` from the structure. The `field` parameter represents one of the frame fields defined in Table 6-2.

6.3.5 Primitives

The primitives referenced in this state diagram are defined in 4.3.1.

6.3.6 State Diagram

The VLC sublayer shall implement the Receive process as defined in the state diagram in Figure 6-4.

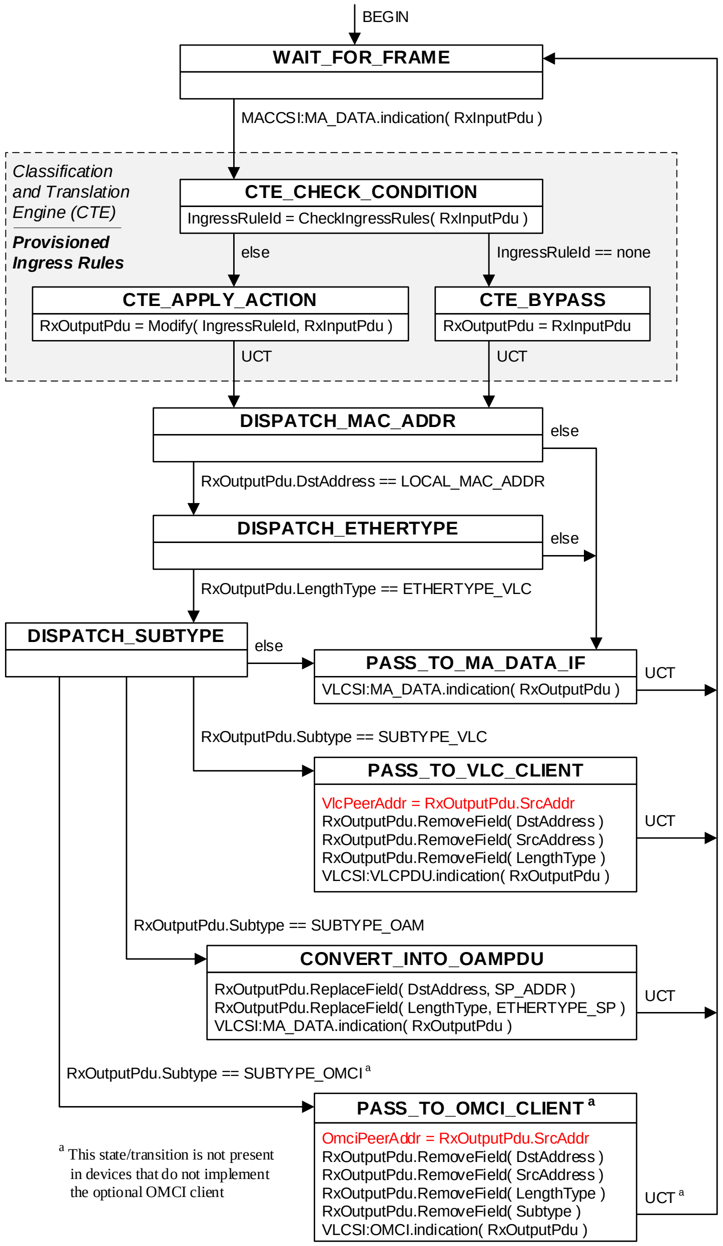
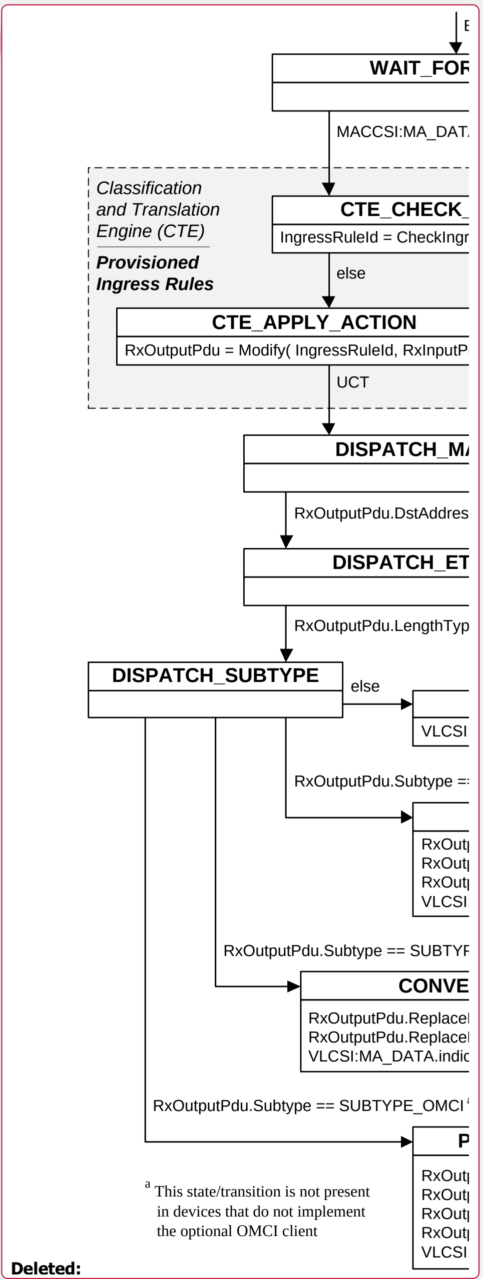

BEGIN
WAIT_FOR_FRAME
MACCSI:MA_DATA.indication(RxInputPdu)
Classification and Translation Engine (CTE)
Provisioned Ingress Rules
CTE_CHECK_CONDITION
IngressRuleId = CheckIngressRules(RxInputPdu)
else
IngressRuleId == none
CTE_APPLY_ACTION
RxOutputPdu = Modify(IngressRuleId, RxInputPdu)
CTE_BYPASS
RxOutputPdu = RxInputPdu
UCT
UCT
DISPATCH_MAC_ADDR
else
RxOutputPdu.DstAddress == LOCAL_MAC_ADDR
DISPATCH_ETHERTYPE
else
RxOutputPdu.LengthType == ETHERTYPE_VLC
DISPATCH_SUBTYPE
else
PASS_TO_MA_DATA_IF
VLCSI:MA_DATA.indication(RxOutputPdu)
UCT
RxOutputPdu.Subtype == SUBTYPE_VLC
PASS_TO_VLC_CLIENT
VlcPeerAddr = RxOutputPdu.SrcAddr
RxOutputPdu.RemoveField(DstAddress)
RxOutputPdu.RemoveField(SrcAddress)
RxOutputPdu.RemoveField(LengthType)
VLCSI:VLCPDU.indication(RxOutputPdu)
UCT
RxOutputPdu.Subtype == SUBTYPE_OAM
CONVERT_INTO_OAMPDU
RxOutputPdu.ReplaceField(DstAddress, SP_ADDR)
RxOutputPdu.ReplaceField(LengthType, ETHERTYPE_SP)
VLCSI:MA_DATA.indication(RxOutputPdu)
UCT
RxOutputPdu.Subtype == SUBTYPE_OMCI a
PASS_TO_OMCI_CLIENT a
OmciPeerAddr = RxOutputPdu.SrcAddr
RxOutputPdu.RemoveField(DstAddress)
RxOutputPdu.RemoveField(SrcAddress)
RxOutputPdu.RemoveField(LengthType)
RxOutputPdu.RemoveField(Subtype)
VLCSI:OMCI.indication(RxOutputPdu)
UCT a
a This state/transition is not present in devices that do not implement the optional OMCI client

Figure 6-1—Receive process state diagram


WAIT_FOR
MACCSI:MA_DAT
Classification and Translation Engine (CTE)
Provisioned Ingress Rules
CTE_CHECK_
IngressRuleId = CheckIngr
else
CTE_APPLY_ACTION
RxOutputPdu = Modify(IngressRuleId, RxInputP
UCT
DISPATCH_M
RxOutputPdu.DstAddres
DISPATCH_ET
RxOutputPdu.LengthTyp
DISPATCH_SUBTYPE
else
VLCSI
RxOutputPdu.Subtype =
RxOut
RxOut
RxOut
VLCSI
RxOutputPdu.Subtype == SUBTYP
CONVE
RxOutputPdu.Replace
RxOutputPdu.Replace
VLCSI:MA_DATA.indic
RxOutputPdu.Subtype == SUBTYPE_OMCI
P
RxOut
RxOut
RxOut
RxOut
VLCSI
a This state/transition is not present in devices that do not implement the optional OMCI client

Deleted:

Deleted: 4

6.4 Transmit path specification

6.4.1 Principles of operation

The transmit path of the VLC sublayer includes the Transmit process. The Transmit process waits for an xPDU to be received from one of the VLCSI interfaces: (*VLCSI:MA_DATA*, *VLCSI:VLCPDU*, or *VLCSI:OMCI*).

If a VLC xPDU is received from the *VLCSI:VLCPDU* interface, it is converted into a VLCPDU with subtype SUBTYPE_VLC (see Table 5-1) by prepeding a VLCPDU header to the VLC xPDU payload. The header cosnsists of the destination address, source address, and Ethertype fields. If a VLCPDU with subtype SUBTYPE_VLC has been previously received from the peer VLC entity, the destination address value is set to the MAC address of that VLC peer. Otherwise, the destination address is set to the default value of NULL_MAC_ADDR.

Deleted: Note that both the destination and the source addresses are equal to the local MAC address assigned to the given port.

If an OMCI xPDU is received from the *VLCSI:OMCI* interface, it is converted into VLCPDU with subtype SUBTYPE_OMCI (see Table 5-1) by prepeding a VLCPDU header to the VLC xPDU payload. The header cosnsists of the destination address, source address, Ethertype, and subtype fields. If a VLCPDU with subtype SUBTYPE_OMCI has been previously received from the peer OMCI entity, the destination address value is set to the MAC address of that OMCI peer. Otherwise, the destination address is set to the default value of NULL_MAC_ADDR.

Deleted: Note that both the destination and the source addresses are equal to the local MAC address assigned to the given port.

After the above modifications, the VLC or OMCI xPDU is formed into a complete frame, which is then processed by the Egress Classification and Translation Engine (CTE). If a match is found, the frame is modified according to the matched rule action. If the frame does not match any rules, it is passed through the CTE block unmodified.

Note that to enter a tunnel, any VLCPDU that contains the default destination address value of NULL_MAC_ADDR require a matching egress CTE rule that, at a minimum, overwrites the NULL_MAC_ADDR value with the MAC address associated with the xPDU destination for the given tunnel. In absence of such rule, a frame that is left with a default destination address value of NULL_MAC_ADDR is discarded by the VLC Transmit process.

Deleted: the VLC xPDU or the OMCI xPDU

Deleted: local MAC address

Deleted: in the VLCPDU destination address field

NOTE – An OAMPDU received from the higher-layer entity (OAM sublayer) via the VLCSI:MA_DATA.request() primitive is not unconditionally converted into VLCPDU by the Transmit process state diagram. However, if there is an egress rule provisioned that matches that OAMPDU, it may get converted into a VLCPDU, as explained in 6.2.

6.4.2 Constants

The constants referenced in this state diagram are defined in 6.3.2.

6.4.3 Variables

DstAddress

This variable is defined in 6.3.3.

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.

LengthType

This variable is defined in 6.3.3.

OmciPeerAddr

This variable is defined in 6.3.3.

SrcAddress

This variable is defined in 6.3.3.

Subtype

This variable is defined in 6.3.3.

TxInputPdu

TYPE: structure containing an Ethernet frame

This variable holds a PDU received from one of the the VLCSI interfaces (*VLCSI:VLCPDU*, *VLCSI:OMCI*, or *VLCSI:MA_DATA*). When received from the *VLCSI:MA_DATA* interface, the TxInputPdu structure contains a complete and properly-formed Ethernet frame. When received from *VLCSI:VLCPDU* or *VLCSI:OMCI* interfaces, the TxInputPdu structure contains a partial frame, that only includes the parameters defined for the respective request() primitive (see 4.3.1).

Additionally, the TxInputPdu structure supports the AddField(field, field_value) method, which adds a field identified by the field and having the value field_value to the structure.

TxOutputPdu

TYPE: structure containing an Ethernet frame

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.

VlcPeerAddr

This variable is defined in 6.3.3.

6.4.4 Functions

CheckEgressRules(input_pdu)

This function returns the identification of an ingress rule that matched the frame contained in TxInputPdu structure. If multiple rules match a frame, the function returns a single identification of any of these rules. The selection criteria is vendor-specific and outside the scope of this standard. If none of the rules matches the frame, a special value none is returned.

IsValidFrame(output_pdu)

This function returns true if the output_pdu structure contains a valid Ethernet frame. Otherwise, false is returned. This function verifies the presence of the DstAddr, SrcAddr, LengthType, and Subtype fileds and that the DstAddr field value is not equal to the default value of NULL_MAC_ADDR.

```
bool IsValidFrame(output_pdu)
{
    return ( exists( output_pdu.DstAddr ) AND
             exists( output_pdu.SrdAddr ) AND
             exists( output_pdu.LengthType ) AND
             output_pdu.DstAddr != NULL_MAC_ADDR );
}
```

The `exists(field)` operator is defined in Table 6-1.

`Modify(rule_id, input_pdu)`

This functions is defined in 6.3.4.

`TxInputPdu.AddField(field, field_value)`

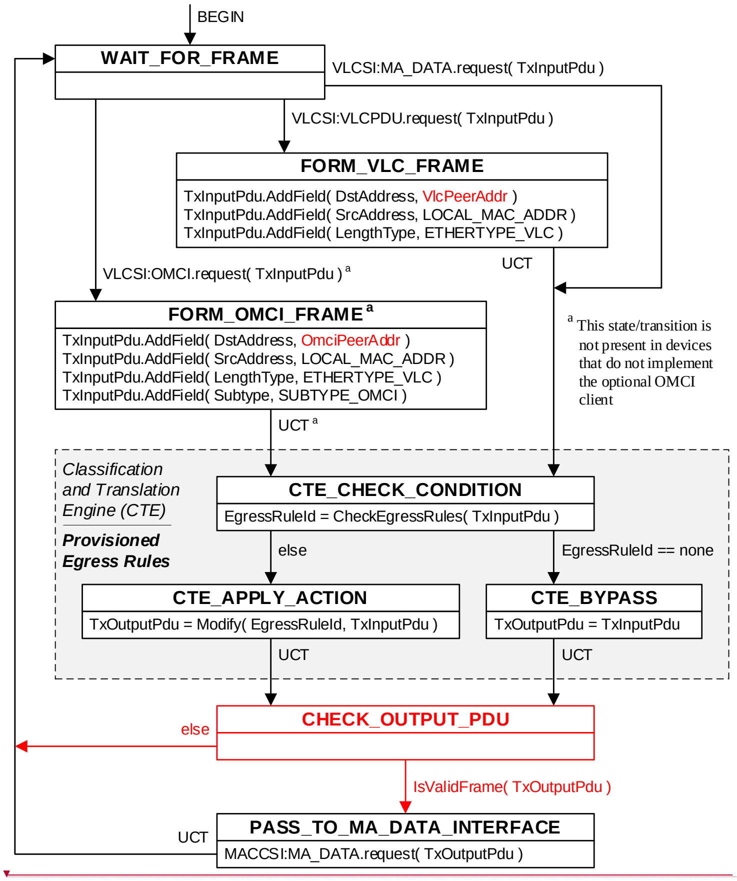
This function is a method associated with `TxInputPdu` structure used in the Transmit process state diagram. This method adds the `field` with the value of `field_value` into the structure. The `field` parameter represents one of the frame fields defined in Table 6-2.

6.4.5 Primitives

The primitives referenced in this state diagram are defined in 4.3.1.

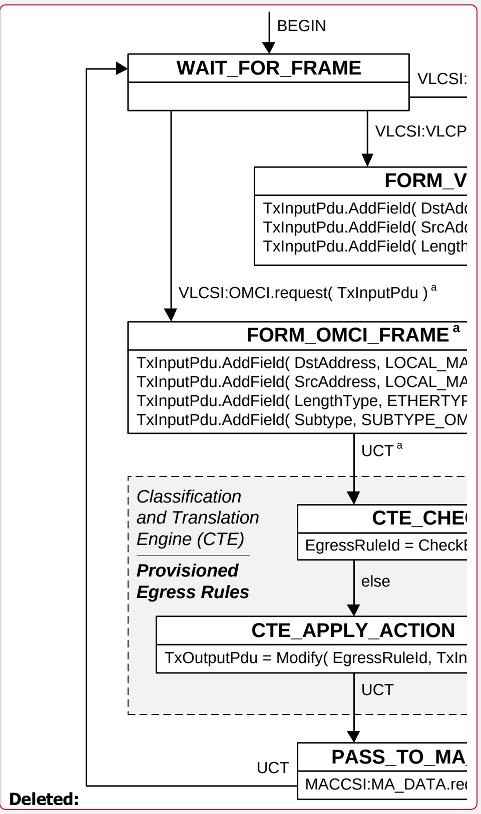
6.4.6 State Diagram

The VLC sublayer shall implement the Transmit process as defined in the state diagram in Figure 6-5.


BEGIN
WAIT_FOR_FRAME
VLCSI:MA_DATA.request(TxInputPdu)
VLCSI:VLCPDU.request(TxInputPdu)
FORM_VLC_FRAME
TxInputPdu.AddField(DstAddress, VlcPeerAddr)
TxInputPdu.AddField(SrcAddress, LOCAL_MAC_ADDR)
TxInputPdu.AddField(LengthType, ETHERTYPE_VLC)
UCT
VLCSI:OMCI.request(TxInputPdu) [a]
FORM_OMCI_FRAME [a]
TxInputPdu.AddField(DstAddress, OmciPeerAddr)
TxInputPdu.AddField(SrcAddress, LOCAL_MAC_ADDR)
TxInputPdu.AddField(LengthType, ETHERTYPE_VLC)
TxInputPdu.AddField(Subtype, SUBTYPE_OMCI)
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
CHECK_OUTPUT_PDU
else
IsValidFrame(TxOutputPdu)
UCT
PASS_TO_MA_DATA_INTERFACE
MACCSI:MA_DATA.request(TxOutputPdu)

[a] This state/transition is not present in devices that do not implement the optional OMCI client

Figure 6-2—Transmit process state diagram


BEGIN
WAIT_FOR_FRAME
VLCSI:
VLCSI:VLCP
FORM_V
TxInputPdu.AddField(DstAdd
TxInputPdu.AddField(SrcAdd
TxInputPdu.AddField(Length
VLCSI:OMCI.request(TxInputPdu) [a]
FORM_OMCI_FRAME [a]
TxInputPdu.AddField(DstAddress, LOCAL_MA
TxInputPdu.AddField(SrcAddress, LOCAL_MA
TxInputPdu.AddField(LengthType, ETHERTYP
TxInputPdu.AddField(Subtype, SUBTYPE_OM
UCT [a]
Classification and Translation Engine (CTE)
Provisioned Egress Rules
CTE_CHE
EgressRuleId = CheckE
else
CTE_APPLY_ACTION
TxOutputPdu = Modify(EgressRuleId, TxIn
UCT
UCT
PASS_TO_MA
MACCSI:MA_DATA.re

Deleted:

Deleted: 5