

1 1.1 Receive path specification

2 **Editorial Note:** This is subclause 6.2

3 1.1.1 Principles of operation

4 The receive path of the VLC sublayer includes the Receive process. The Receive process waits for assertion
5 of the `MACCSI:MA_DATA.indication()`, as defined in **4.3.1.x**.

6 Upon assertion of `MACCSI:MA_DATA.indication()`, the received frame is processed by the ingress
7 Classification and Translation Engine (CTE) and if a matching rule is found, the frame is modified according
8 to the matched rule's action. If the frame does not match any rules, it is passed through the CTE block
9 unmodified.

10 After traversing the ingress CTE block, the frame is dispatched to one of the VLCSI interfaces:
11 (VLCSI:VLCPPDU, VLCSI:OMCI, or VLCSI:MA_DATA). The dispatching decision is based on the values
12 of the MAC destination address, Ethernet Type/Length, and VLC subtype.

13 VLCPDUs with the MAC destination address matching the local MAC address and the VLC subtype equal
14 to VLC_SUBTYPE (see **Error! Reference source not found.**) are modified to match the parameters expected
15 by the VLCSI:VLCPPDU.indication() primitive (see **4.3.1.x**) and the VLC sublayer passes those
16 parameters to the higher-layer entity by asserting the `VLCSI:VLCPPDU.indication()` primitive.

17 VLCPDUs with the MAC destination address matching the local MAC address and the VLC subtype equal
18 to OAM_SUBTYPE (see **Error! Reference source not found.**) are converted into OAMPDUs by the CTE.
19 The resulting OAMPDUs are passed by the VLC sublayer to the higher-layer entity by asserting the
20 `VLCSI:MA_DATA.indication()` primitive.

21 The VLCPDUs with the destination address matching the local MAC address and the VLC subtype equal to
22 OMCI_SUBTYPE (see **Error! Reference source not found.**) are modified to match the parameters expected
23 by the VLCSI:OMCI.indication() primitive (see **Error! Reference source not found.**) and the VLC
24 sublayer passes those parameters to the higher-layer entity by asserting `VLCSI:OMCI.indication()`
25 primitive.

26 All other xPDUs are pass through the CTE unmodified and the VLC sublayer asserts the
27 `VLCSI:MA_DATA.indication()` primitive to pass the unmodified xPDUs to the higher-layer entity
28 where they may be consumed by a local client or bridged to another port.

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

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

35 1.1.2 Constants

36 DST_ADDR

37 This constant identifies a field in a frame, as defined in **Error! Reference source not found.**

Deleted: for a frame to be received on

Deleted: interface (via `MACCSI:MA_DATA.indication()` primitive, ...

Deleted:)

Formatted: Font: (Default) Courier New

Deleted: When a frame is received,

Deleted: it

Deleted: (highlighted in Figure 6-4)

Deleted: t

Deleted:

Deleted: The

Deleted: Table 5-1

Deleted: are passed to the VLCSI:VLCPPDU interface

Deleted: The

Deleted: Table 5-1

Deleted: and are passed to the

Deleted: interface.

Deleted: Table 5-1

Deleted: 4.3.1.4.2

Deleted: and are passed to the VLCSI:OMCI interface

Deleted: passed

Deleted: to the

Deleted: interface. Note that there still may be other local clients

that will intercept/consume these xPDUs at a higher layer.

Deleted: passed

Deleted: to the

Deleted: interface. Note that there still may be other local clients

that will intercept/consume these xPDUs at a higher layer.

Deleted: passed

Deleted: to the

Deleted: interface. Note that there still may be other local clients

that will intercept/consume these xPDUs at a higher layer.

Deleted: passed

Deleted: to the

Deleted: interface. Note that there still may be other local clients

that will intercept/consume these xPDUs at a higher layer.

Deleted: passed

Deleted: to the

Deleted: interface. Note that there still may be other local clients

that will intercept/consume these xPDUs at a higher layer.

Deleted: passed

Deleted: to the

Deleted: interface. Note that there still may be other local clients

that will intercept/consume these xPDUs at a higher layer.

Deleted: passed

Deleted: to the

Deleted: interface. Note that there still may be other local clients

that will intercept/consume these xPDUs at a higher layer.

Deleted: passed

Deleted: to the

Deleted: interface. Note that there still may be other local clients

that will intercept/consume these xPDUs at a higher layer.

Deleted: Table 6-2

1 ETH_TYPE_LEN

2 This constant identifies a field in a frame, as defined in [Error! Reference source not found.](#)

Deleted: Table 6-2

3 LOCAL_MAC_ADDR

4 TYPE: 48-bit MAC address

5 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 MAC address is defined in IEEE Std 802.3, 3.2.3.

8 VALUE: device-specific

9 OMCI_SUBTYPE

10 This constant represents a VLC PDU subtype as defined in [Error! Reference source not found.](#)

Deleted: Table 5-1

11 SP_ADDR

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

14 SP_TYPE

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

17 SRC_ADDR

18 This constant identifies a field in a frame, as defined in [Error! Reference source not found.](#)

Deleted: Table 6-2

19 XPDU SUBTYPE

20 This constant identifies a field in a frame, as defined in [Error! Reference source not found.](#)

Deleted: Table 6-2

21 VLC_ETH_TYPE

22 TYPE: 16-bit Ethernet Type/Length

Deleted: ER

23 This constant holds the Ethernet Type/Length value identifying a frame as a VLC PDU.

Deleted: type

24 VALUE: 0xA8-C8

Deleted: type

Deleted: the

Deleted: s

Deleted:

25 VLC_SUBTYPE

26 This constant represents a VLC PDU subtype as defined in [Error! Reference source not found.](#)

Deleted: Table 5-1

27 1.1.3 Variables

28 IngressRuleId

29 TYPE: 16-bit unsigned integer

30 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.

32 RxInputPdu

33 TYPE: structure

Deleted: containing an Ethernet frame

34 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.

1 RxOutputPdu

2 TYPE: structure

3 This variable holds an Ethernet frame that is the result of processing by the CTE. The fields of this
4 structure correspond to the parameters of the MA_DATA.indication() primitive, as defined in
5 IEEE Std 802.3, 2.3.2.

6 The RxOutputPdu structure supports the RemoveField(field_code) method and the
7 ReplaceField(field_code). The RxOutputPdu structure may contain an incomplete
8 Ethernet frame.

9 **1.1.4 Functions**

10 CheckIngressRules(input_pdu)

11 This function returns the identification of an ingress rule that matched the frame contained in
12 RxInputPdu structure. If multiple rules matched the frame, the function returns an identification
13 of any of these rules. If none of the rules matched the frame, a special value none is returned.

14 Modify(rule_id, input_pdu)

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

17 **1.1.5 Primitives**

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

19 **1.1.6 State Diagram**

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

Deleted: containing an Ethernet frame

Deleted: to be passed to one of the the VLCSI interfaces (VLCSI:VLCPPDU, VLCSI:OMCI, or VLCSI:MA_DATA).

Deleted: Additionally, the

Deleted: , which removes a field identified by the field_code from the structure.

Deleted: Thus, unlike the RxInputPdu structure, the

Deleted: only a partial

Deleted: . The field_code parameter takes values as defined in Table 6-2

Deleted: 4.3.1

Deleted: Figure 6-4

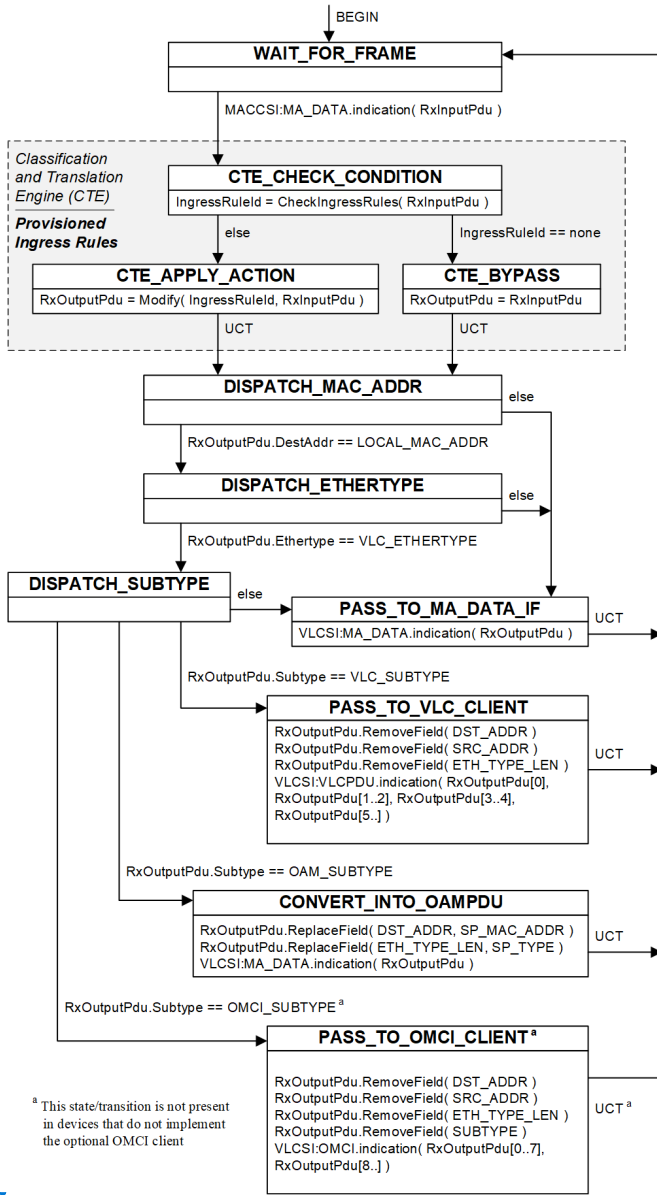
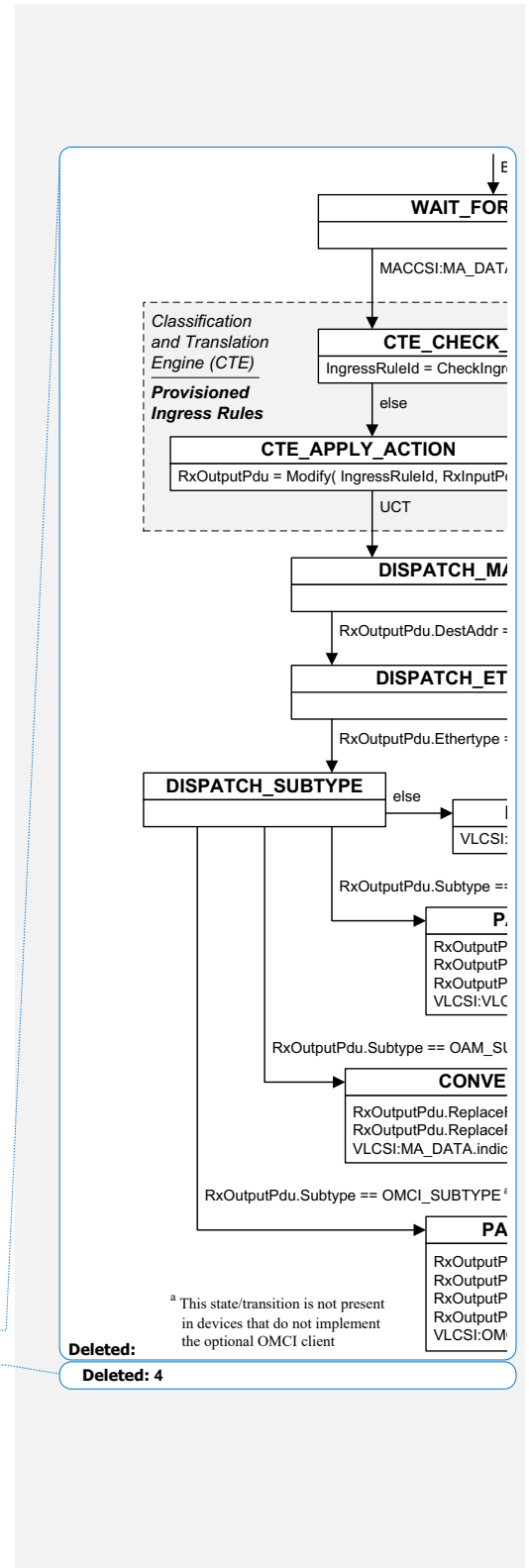


Figure Error! No text of specified style in document. -1- Receive process state diagram



Deleted:
Deleted: 4

1
2
3