1.1 Receive path specification

Editorial Note: This is subclause 6.2

1.1.1 Principles of operation

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

Upon assertion of `MACCSI:MA_DATA.indication()`, the received frame is processed by the ingress Classification and Translation Engine (CTE) and if a matching rule 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.

After traversing the ingress CTE block, 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, Ethernet Type/Length, and VLC subtype.

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

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

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

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

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 destined for a local client is built-in into the Receive process.

1.1.2 Constants

`DST_ADDR`

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

`ETH_TYPE_LEN`

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

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

VALUE: device-specific

`OMCI_SUBTYPE`

This constant represents a VLCPDU subtype as defined in **Error! Reference source not found.**.

`SP_ADDR`

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

`SP_TYPE`

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

`SRC_ADDR`

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

`XPDU_SUBTYPE`

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

`VLC_ETH_TYPE`

TYPE: 16-bit Ethernet Type/Length

This constant holds the Ethernet Type/Length value identifying a frame as a VLCPDU.

VALUE: 0xA8-C8

`VLC_SUBTYPE`

This constant represents a VLCPDU subtype as defined in **Error! Reference source not found.**.

1.1.3 Variables

`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.

`RxInputPdu`

TYPE: structure

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

This variable holds an Ethernet frame that is the result of processing by the CTE. 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.

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

1.1.4 Functions

`CheckIngressRules(input_pdu)`

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

`Modify(rule_id, input_pdu)`

This functions 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.

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

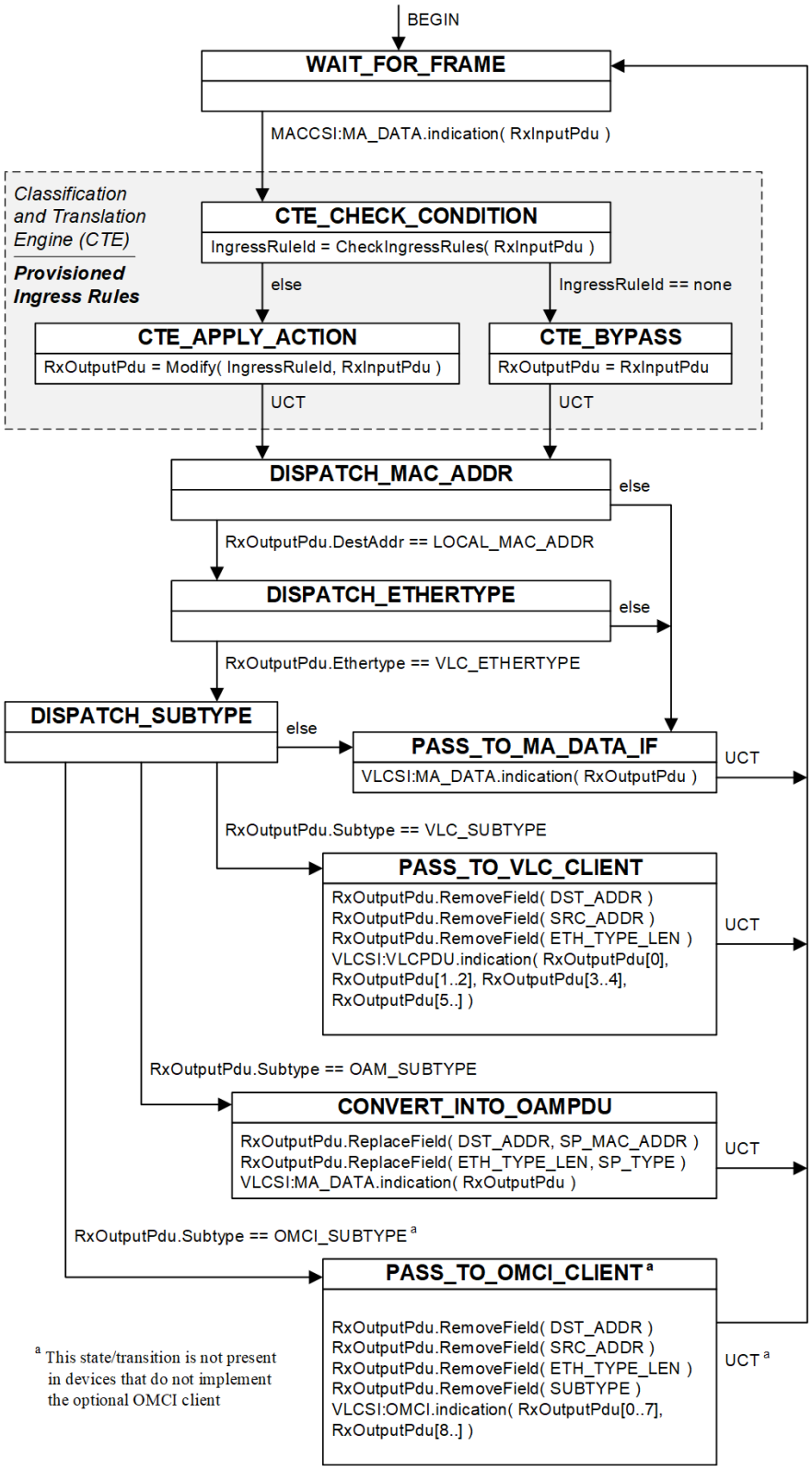

Figure Error! No text of specified style in document.**-1—Receive process state diagram**