1 Virtual Link Control (VLC) Overview and Architecture

Editorial Note: This is clause 4

1.1 Principles of operation

Virtual Link Control (VLC) defines the method of encapsulating various protocol data units (xPDUs) in Ethernet frames with VLC Ethertype (0xA8-C8). An Ethernet frame with VLC Ethertype is called a Virtual Link Control Protocol Data Unit (VLCPDU). That portion of the network path that xPDUs traverse while they are encapsulated as VLCPDUs is referred to as a *tunnel*.

The xPDU-to-VLCPDU and VLCPDU-to-xPDU conversions take place within the VLC sublayer (see 1.2). Both VLC client and VLC sublayer are optional, i.e., in any multi-port device, the VLC sublayer may be implemented in only some ports. Devices that implement the VLC sublayer in at least one of the ports are said to be VLC-aware.

Devices that do not implement VLC sublayer in any of the ports are called VLC-unaware. VLC-unaware devices are able to relay VLCPDUs as generic Ethernet frames using existing L2 forwarding mechanisms but are unable to consume or generate VLCPDUs.

All VLCPDUs except the *VLC_CONFIG* VLCPDUs carry tunneling payloads associated with specific protocols (xPDU). Any payload-carrying VLCPDU that is consumed by a device is first converted into its native xPDU format and then passed to a specific client associated with that xPDU protocol type. Correspondingly, any payload-carrying VLCPDU that is generated by a device originates in a protocol-specific client as xPDU and is then converted into VLCPDU within the VLC sublayer.

A device port where xPDUs are converted into VLCPDUs (within the VLC sublayer) is referred to as *VLC entrance point* and a port where the opposite conversion takes place is referred to as *VLC exit point*.

1.1.1 VLC discovery protocol

The tunnel entrance and exit points may be pre-configured or provisioned via *VLC_CONFIG* VLCPDUs based on known network topology and L2 device addresses. An automatic VLC discovery protocol is out-of-scope for this revision of the standard.

1.2 VLC sublayer

VLC functionality is confined to the VLC sublayer. **Error! Reference source not found.** depicts architectural positioning of the VLC sublayer, which is a client of the MAC Control sublayer (see IEEE Std 802.3, Clause 31). The VLC Sublayer functionality is fully specified in Clause 6.

2 VLC sublayer

Editorial Note: This is clause 6

The VLC Sublayer is where xPDU-to-VLCPDU and VLCPDU-to-xPDU conversions take place. The internal functional structure of the VLC Sublayer is shown in Figure 6-1.

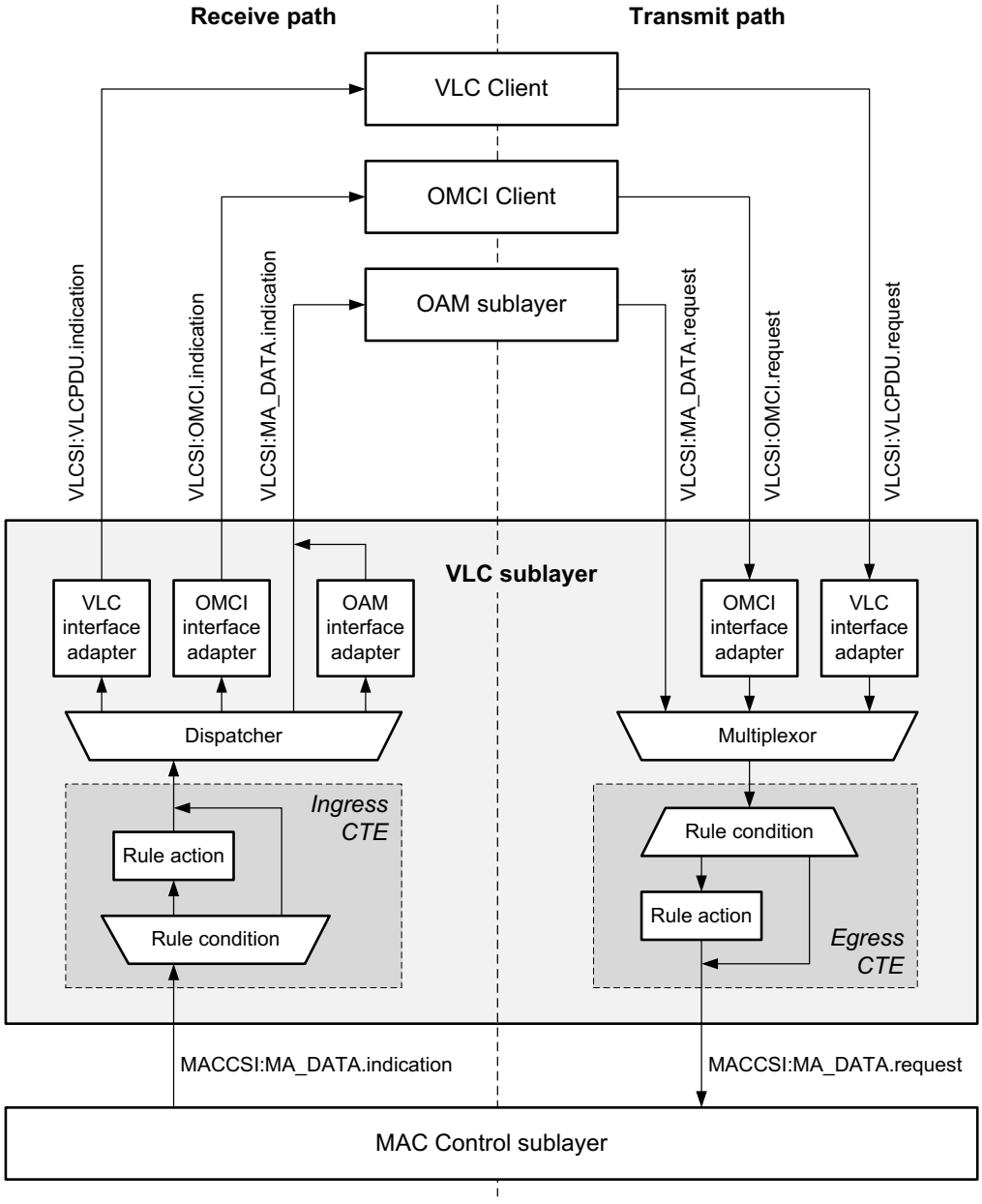

Figure 2-1—VLC sublayer functional block diagram

The VLC sublayer includes a set of interface adapters and the Classification and Translation Engine (CTE). Together these functional blocks convert xPDUs into VLCPDUs and vice versa. The CTE behavior is governed by a set of rules that are either statically configured or dynamically provisioned by the NMS (see 2.1).

The VLC sublayer also includes the conceptual Dispatcher block and Multiplexor block. The Dispatcher is responsible for distributing xPDUs processed in the Ingress CTE to the appropriate higher-layer block. The Multiplexor is responsible for multiplexing xPDUs received from higher-layer blocks into the egress CTE.

The interface adapter blocks and the Multiplexor and Dispatcher blocks are not specified separately in this standard, but the equivalent functionality is incorporated into the transmit path specification in 6.3 and the receive path specification in 6.2.

The VLC sublayer provides a service interface to the OAM sublayer, the VLC client, and may provide service interface to other L2 protocol-specific clients. The only messages that are passed to and received from the VLC client are the VLC configuration messages (see *VLC_CONFIG* VLCPDU in **Error! Reference source not found.**).

2.1 VLC Classification and Translation Engine

The function of the VLC Classification and Translation Engine (CTE) is to classify frames by certain criteria and to perform specific modification on the frames that match the criteria. The classification criteria together with the associated modification action comprise an entity called a *rule*. The concept of a rule is similar to that defined in IEEE 1904.1, 6.5.2.1.

By matching frames to specific rules, the CTE is able to translate VLCPDUs into xPDUs (i.e., into frames with different Ethertype values) and vice versa.

There are separate CTE instances in the transmit path and in the receive path of each physical or virtual port. The CTE located in the receive path is called *Ingress CTE* and the CTE located in the transmit path is called *Egress CTE* (see Figure 2-1). Fundamentally, a CTE instance is simply a table that stores multiple rules. Some of the rules are statically pre-configured (i.e., available and active at all times); other rules are dynamically added/deleted by NMS when tunnels are established or destroyed.