6 UMT sublayer

6.1 UMT Classification and Translation Engine

The function of the UMT 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 UMTPDUs into xPDUs (i.e., into frames with different Ethertype values) and vice versa.

Moved down [1]: A frame that does not match any CTE rules traverses the UMT sublayer without any modifications.

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 **Error! Reference source not found.**). 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.

Deleted: Figure 6-1

6.1.1 CTE rule structure

Deleted: ¶

6.1.1.1 CTE rule processing

The Ingress CTE and Egress CTE each maintain a rules table. Rules can be installed into or removed from the CTE tables by local configuration mechanisms or by the UMT Configuration protocol (See Clause 7).

Upon receipt of the UMTSI:MA_DATA.request primitive or any other request primitive from a higher-layer client, the CTE compares the received parameters to the match conditions of the rules in the egress CTE rules table. Each rule is tested in order of precedence. Only the first rule to match is executed. In the case that two or more rules have equal precedence, then the most-specific matching rule is executed. A frame that does not match a rule is not modified.

Moved (insertion) [1]

Deleted: A

Deleted: any

Deleted: CTE

Deleted: s

Deleted:

Deleted: traverses the UMT sublayer without any modifications

Deleted: ¶

The result of rules processing must contain the parameters necessary to invoke the MACCSI:MA_DATA.request primitive. If the result does not meet these criteria, then the xPDU is dropped.

Upon completion of rule processing, the Egress CTE asserts the MACCSI:MA_DATA.request primitive with the result of the rules processing.

Upon receipt of the MACCSI:MA_DATA.indication primitive, the CTE compares the received parameters to the match conditions of the rules in the egress CTE rules table. Each rule is tested in order of precedence. Only the first rule to match is executed. In the case that two or more rules have equal precedence, then the most-specific matching rule is executed. A frame that does not match a rule is not modified.

Upon completion of rule processing, the Egress CTE asserts the indication primitive associated with the result of the rules processing. The result of rules processing must contain the parameters necessary to invoke the specified primitive. If the result does not meet these criteria, then the xPDU is dropped.

A condition may compare a particular field in a frame against a provisioned value, test for existence of a field, or unconditionally return “true” or “false”. A condition consists of a comparison operator and one or two operands. Supported comparison operators are listed in 6.1.1.1.1. An operand may be a numeric value or a code representing a specific field in the frame’s header. Supported field codes are listed in 6.1.1.1.1. The same field may be used in multiple comparisons (either in different rules or in different conditions of the same rule). The results of all conditions provisioned for a given rule are logically ANDed together to determine whether the rule is a match. If all conditions in a rule evaluate to “true”, the rule is considered to match the frame. A rule match causes all the actions associated with the rule to be applied to the frame.

Deleted: header

Deleted: 6.1.1.1.2

6.1.1.1.1 Comparison operators

6.1.1.1.2 Classification fields

The CTE comparison operation elements recognize the fields shown in Table 6-1. Note that field codes listed below represent unique identifiers of various fields accessible to the CTE rules. The field codes are shown in all capital letters as opposed to the field names, which are shown as a mixture of capital and lowercase letters.

Deleted: Table 6-2

Table 6-1—L2 classification fields

Deleted: 2

FIELD_CODE	Numeric Code	Field size (bits)	Description
DST_ADDR	0x01	48	Outermost MAC Destination Address.
SRC_ADDR	0x02	48	Outermost MAC Source Address.
ETH_TYPE_LEN	0x03	16	Outermost Ethernet Type/Length field, per IEEE Std 802.3, 3.1.1
PRIMITIVE	0x020	1	
VLAN0	0x04	32	*Outermost VLAN tag*. This parameter corresponds to the first VLAN tag following the SRC_ADDR field. If no VLAN tags follow the SRC_ADDR field, then the VLAN0 field does not exist.
VLAN0_TPID	0x05	16	*Tag Protocol Identifier* of the VLAN0.
VLAN0_VID	0x06	12	*VLAN Identifier* of the VLAN0.
VLAN1	0x07	32	*Innermost VLAN tag*. This parameter corresponds to the VLAN tag that follows the outermost tag VLAN0. If no VLAN tags follow the VLAN0 field, then the VLAN1 field does not exist.
VLAN1_TPID	0x08	16	*Tag Protocol Identifier* of the VLAN1.
VLAN1_VID	0x09	12	*VLAN Identifier* of the VLAN1.
UMT_DST_ADDR	0x11	48	*UMTPDU MAC Destination Address*. In UMTPDUs, this field code is eqivalent to DST_ADDR. In other (non-UMT) PDU types, this field does not exist.
UMT_SRC_ADDR	0x12	48	*UMTPDU MAC Source Address*. In UMTPDUs, this field code is eqivalent to SRC_ADDR. In other (non-UMT) PDU types, this field does not exist.
UMT_ETH_TYPE	0x13	16	*UMT Ethernet Type*. In UMTPDUs, this field code is eqivalent to ETH_TYPE_LENGTH. In other (non-UMT) PDU types, this field does not exist.
UMT_VLAN0	0x14	32	*UMTPDU Outermost VLAN tag*. In UMTPDUs, this field code is eqivalent to VLAN0. In other (non-UMT) PDU types, this field does not exist.
UMT_VLAN0_TPID	0x15	16	*Tag Protocol Identifier* of the UMT_VLAN0. In UMTPDUs, this field code is eqivalent to VLAN0_TPID. In other (non-UMT) PDU types, this field does not exist.

FIELD_CODE	Numeric Code	Field size (bits)	Description
UMT_VLAN0_VID	0x16	12	*VLAN Identifier* of the UMT_VLAN0. In UMTPDUs, this field code is eqivalent to VLAN0_VID. In other (non-UMT) PDU types, this field does not exist.
UMT_VLAN1	0x17	32	*UMTPDU Innermost VLAN tag*. In UMTPDUs, this field code is eqivalent to VLAN1. In other (non-UMT) PDU types, this field does not exist.
UMT_VLAN1_TPID	0x18	16	*Tag Protocol Identifier* of the UMT_VLAN1. In UMTPDUs, this field code is eqivalent to VLAN1_TPID. In other (non-UMT) PDU types, this field does not exist.
UMT_VLAN1_VID	0x19	12	*VLAN Identifier* of the UMT_VLAN1. In UMTPDUs, this field code is eqivalent to VLAN1_VID. In other (non-UMT) PDU types, this field does not exist.
UMT_SUBTYPE	0x1A	8	*UMT Subtype field*. This field exists in UMTPDUs only, where it is located immediately after the *UMT_ETH_TYPE* field.
XPDU_DST_ADDR	0x21	48	*xPDU MAC Destination Address*. In xPDUs (non-UMT types), this field code is eqivalent to DST_ADDR. In UMTPDUs, this field does not exist.
XPDU_SRC_ADDR	0x22	48	*xPDU MAC Source Address*. In xPDUs (non-UMT types), this field code is eqivalent to SRC_ADDR. In UMTPDUs, this field does not exist.
XPDU_ETH_TYPE	0x23	16	*xPDU Ethernet Type*. In xPDUs (non-UMT types), this field code is eqivalent to ETH_TYPE_LENGTH. In UMTPDUs, this field does not exist.
XPDU_VLAN0	0x24	32	*xPDU Outermost VLAN tag*. In xPDUs (non-UMT types), this field code is eqivalent to VLAN0. In UMTPDUs, this field does not exist.
XPDU_VLAN0_TPID	0x25	16	*Tag Protocol Identifier* of the XPDU_VLAN0. In xPDUs (non-UMT types), this field code is eqivalent to VLAN0_TPID. In UMTPDUs, this field does not exist.
XPDU_VLAN0_VID	0x26	12	*VLAN Identifier* of the XPDU_VLAN0. In xPDUs (non-UMT types), this field code is eqivalent to VLAN0_VID. In UMTPDUs, this field does not exist.
XPDU_VLAN1	0x27	32	*xPDU Innermost VLAN tag*. In xPDUs (non-UMT types), this field code is eqivalent to VLAN1. In UMTPDUs, this field does not exist.

FIELD_CODE	Numeric Code	Field size (bits)	Description
XPDU_VLAN1_TPID	0x28	16	*Tag Protocol Identifier* of the XPDU_VLAN1. In xPDUs (non-UMT types), this field code is eqivalent to VLAN1_TPID. In UMTPDUs, this field does not exist.
XPDU_VLAN1_VID	0x29	12	*VLAN Identifier* of the XPDU_VLAN1. In xPDUs (non-UMT types), this field code is eqivalent to VLAN1_VID. In UMTPDUs, this field does not exist.

Formatted: Normal

6.1.2 CTE rule categories

6.2 Receive path specification

Deleted: ¶

6.3 Transmit path specification

6.4 CTE rules involving operations on the VLAN tags

7 UMT configuration

Deleted: ¶

7.1 Configuration UMTPDU

7.2 CTE rule TLV structure

The structure of a CTE rule TLV is shown in Table 7-1. Each *UMT_CONFIG* UMTPDU shall contain at least one CTE rule TLV.

Deleted: Table 7-3

Table 7-1—CTE rule TLV structure

Deleted: 3

Field Size (octets)	Field Name	Value	Description
1	*Type*	0xC0	Type code identifying the condition-encoding TLV
		0xAC	Type code identifying the action-encoding TLV
		0x00	Type code indicating that there are no more TLVs to process. The Length field and other fields (if present) are ignored. The TLV with Type = 0x00 shall be the last TLV in every *UMT_CONFIG* UMTPDU and it may be the only TLV in the *UMT_CONFIG* UMTPDU.
1	*Length*	*L*+4	The *Length* field encompasses the entire TLV, including the *Type* and *Length* fields. A TLV with length of 0x00 or 0x01 is invalid, and on reception, should be treated as TLV with Type 0x00.
1	*Table_Spec*	*per Table X-X*	The *Priority* field specifies the order in which the rule should be processed when a frame matches more than one rule.
1	*Priority*	*Varies*	The *Priority* field specifies the order in which the rule should be processed when a frame matches more than one rule.
1	*Operation*	per Table	Comparison operator code, if the TLV *Type* = 0xC0
		per **Error! Reference source not found.**	Action code, if the TLV *Type* = 0xAC
1	*FieldCode*	per Table 6-1	Identifies a field to be used in a comparison, or to be modified by an action.
L	*Value*	various	The value to be used in a comparison or by an Add/Change action. Some TLVs may omit this field.

Deleted: Table 6-1

Deleted: Table 6-3

Deleted: Table 6-2

NOTE—Fields *Operation* and *FieldCode* are present in all TLVs, even if they are not used. When these fields are not used, they are set to the value of zero.

Table X-X —Table Specification Values for the CTE Rules

CTE Table Instance	Numeric Code	Description
Ingress	0x01	Specifies that the rule is installed in the ingress CTE table
Egress	0x02	Specifies that the rule is installed in the egress CTE table.

Form
Width

Form