6 MAC Client reference model

6.1 Introduction

6.2 MAC Client reference model definition

6.3 EPON Service Path definition

6.4 Frame flow through reference model

6.5 Blocks in the reference model

6.5.1 Input

6.5.2 Classifier

6.5.2.1 Classification element (rule) structure

6.5.2.1.1 Classification fields

The Classifier comparison operation elements recognize the following fields:

a) Port~~/~~ (by index)

~~a)~~b) LLID (by index or value)

~~b)~~c) L2 Fields (see Table 6-1)

~~c)~~d) L3 Fields (see Table 6-2)

~~d)~~e) L4 Fields (see Table 6-3)

~~e)~~f) Any arbitrary frame bit fields accessible through a flexible access mechanism (see 6.5.2.1.1.1)

Note that field codes listed below represent unique identifiers of various fields accessible to the Classifier. 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.

Table 6-1—L2 classification fields

FIELD_CODE	Field size (bits)	Description
DA	48	*Outermost MAC Destination Address*. This field corresponds to — C_DA in the case of frames with no Backbone Service Instance encapsulation, or — B_DA in the case of frames with Backbone Service Instance encapsulation.
SA	48	*Outermost MAC Source Address*. This field corresponds to — C_SA in the case of frames with no Backbone Service Instance encapsulation, or — B_SA in the case of frames with Backbone Service Instance encapsulation.
ETYPE_LEN	16	*Ethernet Type/Length* field, per IEEE Std 802.3, 3.1.1

FIELD_CODE	Field size (bits)	Description
VLAN0	32	*Outermost VLAN tag*. This field corresponds to the first VLAN tag following the C_SA field. If no VLAN tags follow the C_SA field, then the VLAN0 field does not exist.
VLAN0_TPID	16	*Tag Protocol Identifier* of the VLAN0
VLAN0_PCP	3	*Priority Code Point* of the VLAN0
VLAN0_IND	1	*VLAN Tag Indicator*. This field corresponds to — C_CFI if the VLAN0 field is a C_TAG, or — S_DEI if the VLAN0 field is an S_TAG.
VLAN0_VID	12	*VLAN Identifier* of the VLAN0
VLAN1	32	*Innermost VLAN tag*. This field 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	16	*Tag Protocol Identifier* of the VLAN1
VLAN1_PCP	3	*Priority Code Point* of the VLAN1
VLAN1_IND	1	*VLAN Tag Indicator*. This field corresponds to — C_CFI if the VLAN1 field is C_TAG, or — S_DEI if the VLAN1 field is an S_TAG.
VLAN1_VID	12	*VLAN Identifier* of the VLAN1
C_DA	48	*Customer MAC Destination Address* for frames with or without Backbone Service Instance encapsulation, per IEEE Std 802.3, 3.1.1
C_SA	48	*Customer MAC Source Address* for frames with or without Backbone Service Instance encapsulation, per IEEE Std 802.3, 3.1.1
C_TAG	32	*Customer VLAN Tag* (C-component), comprising `C_TPID`, `C_PCP`, `C_CFI`, and `C_VID` fields, per IEEE Std 802.1Q, 9.5 and 9.6 [a]
C_TPID	16	*Tag Protocol Identifier* of the Customer VLAN tag (C_TAG, C-component), per IEEE Std 802.1Q, 9.5
C_PCP	3	*Priority Code Point* of the Customer VLAN tag (C_TAG, C-component), per IEEE Std 802.1Q, 9.6
C_CFI	1	*Canonical Format Indicator* of the Customer VLAN tag (C_TAG, C-component), per IEEE Std 802.1Q, 9.6
C_VID	12	*VLAN Identifier* of the Customer VLAN tag (C_TAG, C-component), per IEEE Std 802.1Q, 9.6
S_TAG	32	*Service VLAN Tag* (S-component), comprising `S_TPID`, `S_PCP`, `S_DEI`, and `S_VID` fields, per IEEE Std 802.1Q, 9.5 and 9.6 [a]
S_TPID	16	*Tag Protocol Identifier* of the Service VLAN tag (S_TAG, S-component), per IEEE Std 802.1Q, 9.5
S_PCP	3	*Priority Code Point* of the Service VLAN tag (S_TAG, S-component), per IEEE Std 802.1Q, 9.6
S_DEI	1	*Drop Eligible Indicator* of the Service VLAN tag (S_TAG, S-component), per IEEE Std 802.1Q, 9.6
S_VID	12	*VLAN Identifier* of the Service VLAN tag (S_TAG, S-component), per IEEE Std 802.1Q, 9.6
B_DA	48	*Backbone MAC Destination Address* for frames with Backbone Service Instance encapsulation
B_SA	48	*Backbone MAC Source Address* for frames with Backbone Service Instance encapsulation
B_TAG	32	*Backbone VLAN tag* (B-component), comprising `B_TPID`, `B_PCP`, `B_DEI`, and `B_VID` fields, per IEEE Std 802.1Q, 9.5 and 9.6 [a]
B_TPID	16	*Tag Protocol Identifier* of the Backbone VLAN tag (B_TAG, B-component), per IEEE Std 802.1Q, 9.5
B_PCP	3	*Priority Code Point* of the Backbone VLAN tag (B_TAG, B-component), per IEEE Std 802.1Q, 9.7

FIELD_CODE	Field size (bits)	Description
B_DEI	1	*Drop Eligible Indicator* of the Backbone VLAN tag (B_TAG, B-component), per IEEE Std 802.1Q, 9.7
B_VID	12	*VLAN Identifier* of the Backbone VLAN tag (B_TAG, B-component), per IEEE Std 802.1Q, 9.7
I_HEADER	144	*Backbone Service Instance Header*, comprising `I_TAG`, `B_DA`, and `B_SA` fields
I_TAG	48	*Backbone Service Instance Tag* (I-component), comprising `I_TPID`, `I_PCP`, `I_DEI`, `I_UCA`, `I_RES`, and `I_SID` fields, per IEEE Std 802.1Q, 9.7 [a]
I_TPID	16	*Tag Protocol Identifier* of the Backbone Service Instance tag (I_TAG, I-component), per IEEE Std 802.1Q, 9.5
I_PCP	3	*Priority Code Point* of the Backbone Service Instance tag (I_TAG, I-component), per IEEE Std 802.1Q, 9.7
I_DEI	1	*Drop Eligible Indicator* of the Backbone Service Instance tag (I_TAG, I-component), per IEEE Std 802.1Q, 9.7
I_UCA	1	*Use Customer Addresses* indicator of the Backbone Service Instance tag (I_TAG, I-component), per IEEE Std 802.1Q, 9.7
I_RES	2	*Reserved* bits of Backbone Service Instance tag (I_TAG, I-component), per IEEE Std 802.1Q, 9.7
I_SID	24	*Backbone Service Instance Identifier*, per IEEE Std 802.1Q, 9.7
MPLS_LSE	32	*MPLS Label Stacking Entry*, as defined in IETF RFC 5462
LINK_INDEX	8	Index of a LLID within the given ONU

[a] Presence of this field is determined by detecting a specific TPID value, as set by the network operator. Multiple TPID values may be specified to identify this field. The default TPID values are as specified in IEEE Std 802.1Q.

Table 6-2—L3 classification fields

FIELD_CODE	Field size (bits)	Description
IP_VERSION	4	*IP Version Field*, as defined in IETF RFC 791, section 3.1, for IPv4 and IETF RFC 2460, section 6, for IPv6
IPv4_HEADER	160	*IPv4 Header*, as defined in IETF RFC 791, section 3.1 (minimum size without options)
IPv4_TTL	8	*IPv4 Time To Live*, as defined in IETF RFC 791, section 3.1
IPv4_DA	32	*IPv4 Destination Address*, as defined in IETF RFC 791, section 3.1
IPv4_SA	32	*IPv4 Source Address*, as defined in IETF RFC 791, section 3.1
IPv6_HEADER	320	*IPv6 Header*, as defined in IETF RFC 2460, section 3 (minimum size without extensions)
IPv6_HOP_LIMIT	20	*IPv6 Hop Limit*, as defined in IETF RFC 2460, section 6
IPv6_DA	128	*IPv6 Destination Address*, as defined in IETF RFC 2460, section 3
IPv6_SA	128	*IPv6 Source Address*, as defined in IETF RFC 2460, section 3
IPv4_TOS	8	*IPv4 Type of Service*, as defined in IETF RFC 791, section 3.1
IPv4_DSCP	6	*IPv4 Differentiated Services Code Point*, as defined in IETF RFC 2474, section 3
IPv4_PROTOCOL	8	*IPv4 Protocol*, as defined in IETF RFC 791, section 3.1
IPv6_NEXT_HEADER	8	*IPv6 Next Header*, as defined in IETF RFC 2460, section 4
IPv6_TC	8	*IPv6 Traffic Class*, as defined in IETF RFC 2460, section 7
IPv6_DSCP	6	*IPv6 Differentiated Services Code Point*, as defined in IETF RFC 2474, section 3

FIELD_CODE	Field size (bits)	Description
IPv6_FLOWLABEL	20	*IPv6 Flow Label*, as defined in IETF RFC 2460, section 6
ICMP_HEADER	64	*ICMP Header*, as defined in IETF RFC 792 for ICMPv4 and IETF RFC 4443 for ICMPv6
IGMP_HEADER	128	*IGMP Header*, as defined in IETF RFC 3376, section 4 (without options)
IGMP_TYPE	8	*IGMP frame*, `Type` field. This field corresponds to — `Version` & `Type` fields, as specified in IETF RFC 1112, Appendix I, or — `Type` field, as specified in IETF RFC 2236, section 2, or — `Type` field, as specified in IETF RFC 3376, section 4.1.
MLD_TYPE	8	*MLD frame*, `Type` field, as specified in IETF RFC 2710, section 3, for MLDv1 and IETF RFC 3810, section 5, for MLDv2

Table 6-3—L4 classification fields

FIELD_CODE	Field size (bits)	Description
TCP_HEADER	160	*TCP Header* (minimum size without options) as specified in IETF RFC 793
UDP_HEADER	64	*UDP Header* as specified in IETF RFC 768
TCP_UDP_SP	16	*TCP/UDP Source Port*, as specified in IETF RFC 793, section 3.1, (TCP) or IETF RFC 768 (UDP)
TCP_UDP DP	16	*TCP/UDP Destination Port*, as specified in IETF RFC 793, section 3.1, (TCP) or IETF RFC 768 (UDP)

7 Connectivity configurations

8 Service performance and QoS guarantees

9 Service availability

10 Power saving

11 Security-oriented mechanisms

12 Discovery and maintenance

13 Extended OAM for EPON

14 Management entities

14.1 Introduction

14.2 Management entities for non-SNMP-optimized eOAM profile

14.3 Management entities for SNMP-optimized eOAM profile

14.4 Management entities for DPoE eOAM profile

14.4.1 Branch 0xD6 “identification”

14.4.2 Branch 0x07 “basic attributes”

14.4.3 Branch 0xD7 “extended attributes”

14.4.3.1 ONU management

14.4.3.2 Bridging

14.4.3.3 Statistics and counters

14.4.3.4 Alarms

14.4.3.5 Encryption

14.4.3.6 Frame processing

14.4.3.6.1 Attribute *aRuleSetConfig* (0xD7/0x05-01)

This attribute represents the current configuration of the rule set associated with the given element (as identified by the *Object Context* TLV).

NOTE—The Classifier rule model used by this profile differs from the model described in 6.5.2.1 in the following aspects:

- — All rules configured on the ONU are verified for each frame, where any frame may match multiple rules. The frame processing does not stop on the first matched rule, as described in 6.5.2.1.
- — Results of multiple rules configured on the ONU and verified to match the given frame are applied to the given frame in order of precedence. Consequently, results associated with higher-priority rules can override partially or completely results associated with lower-priority rules.

This attribute consists of the following sub-attributes: *sPrecedence*, *sClauseCount*, at least one instance of *sClause*, *sResultCount*, and at least one instance of *sResult*. These sub-attributes are defined below:

Sub-attribute *aRuleSetConfig.sPrecedence*

Syntax: Unsigned integer
Range: 0x00 to 0xFF
Remote access: Read/Write
Description: This sub-attribute indicates the precedence of the given classification rule. The lower value indicates the higher precedence.

Sub-attribute *aRuleSetConfig.sClauseCount*

Syntax: Unsigned integer
Range: 0x00 to 0xFF

Remote access: Read/Write
Description: This sub-attribute indicates the total number of clauses configured for the given rule.

Sub-attribute *aRuleSetConfig.sClause[sClauseCount]*
Syntax: Structure
Range: 0x00 to 0xFF
Remote access: Read/Write
Description: This sub-attribute represents a single clause configured for the given rule. The *sClause* sub-attribute is itself a compound sub-attribute that consists of multiple sub-attributes. It is further defined in 14.4.3.6.1.1.

Sub-attribute *aRuleSetConfig.sResultCount*
Syntax: Unsigned integer
Range: 0x00 to 0xFF
Remote access: Read/Write
Description: This sub-attribute indicates the total number of results configured for the given rule.

Sub-attribute *aRuleSetConfig.sResult[sResultCount]*
Syntax: Structure
Range: 0x00 to 0xFF
Remote access: Read/Write
Description: This sub-attribute represents a single result (i.e., an action to be performed on a frame) configured for the given rule. The *sResult* sub-attribute is itself a compound sub-attribute that consists of multiple sub-attributes. It is further defined in 14.4.3.6.1.2.

14.4.3.6.1.1 *aRuleSetConfig.sClause* sub-attribute

This sub-attribute represents one of the clauses used to construct a fully functional frame processing rule. A frame processing rule shall contain at least one *sClause* sub-attribute. All *sClause* sub-attributes for the given frame processing rule are evaluated, and the individual logical results are ANDed to determine the match condition.

This sub-attribute comprises the following, second-level sub-attributes: *sFieldCode*, *sFieldInstance*, *sMaskMsb*, *sMaskLsb*, *sOperator*, and *sMatchVal*.

Sub-attribute *aRuleSetConfig.sClause.sFieldCode*:
Syntax: Enumeration
Remote access: Read/Write
Description: This sub-attribute indicates the field of the frame header used for matching by this instance of *sClause* sub-attribute. The following values are defined:

`LINK_INDEX`:	local logical link index [a]
`DA`:	*Outermost MAC Destination Address* field [b]
`SA`:	*Outermost MAC Source Address* field [b]
`ETYPE_LEN`:	*Ethernet Type/Length* field [b]
`B_DA`:	*Backbone MAC Destination Address* field [b]
`B_SA`:	*Backbone MAC Source Address* field [b]
`I_TAG`:	*Backbone Service Instance Tag* field [b]
`S_TAG`:	*Service VLAN Tag* field [b,e]
`C_TAG`:	*Customer VLAN Tag* field [b,e]
`MPLS_LSE` :	MPLS header [e]

IP_TOS_TC:	depending on the version of IP header present in the frame, either *IPv4 Type of Service* [c] (IPv4_TOS) field or *IPv6 Traffic Class* [c] (IPv6_TC) field [g]
IP_TTL_HL:	depending on the version of IP header present in the frame, either *IPv4 Time-to-Live* [c] (IPv4_TTL) field or *IPv6 Hop Limit* [c] (IPv6_HL) field [g]
IP_PT:	depending on the version of IP header present in the frame, either *IPv4 Protocol Type* [c] (IPv4_PROTOCOL) field or the last Next Header field in the chain of Next Header fields present in the IPv6 extension headers [g]
IPv4_DA:	*IPv4 Destination Address* field [c]
IPv6_DA:	*IPv6 Destination Address* field [c]
IPv4_SA:	*IPv4 Source Address* field [c]
IPv6_SA:	*IPv6 Source Address* field [c]
IPv6_NEXT_HEADER:	*IPv6 Next Header* field [c,f]
IPv6_FLOWLABEL:	*IPv6 Flow Label* field [c]
TCP_UDP_SP:	*TCP/UDP Source Port* field [d]
TCP_UDP_DP:	*TCP/UDP Destination Port* field [d]
B_TAG:	*B-Tag* field [b]
CUST_0:	custom field 0
CUST_1:	custom field 1
CUST_2:	custom field 2
CUST_3:	custom field 3
CUST_4:	custom field 4
CUST_5:	custom field 5
CUST_6:	custom field 6
CUST_7:	custom field 7
LLID_VALUE:	LLID Value [h]

[a] The local logical link index represents the local index of the logical link instantiated on the C-ONU. For example, for a C-ONU supporting 8 L-ONUs, the value of local logical link index ranges from 0 to 7. In this way, the local logical link index has only local, C-ONU–specific meaning. The local logical link index represents the order of registration of the L-ONU. The L-ONUs are registered in the order of increasing numerical value of their MAC addresses.

[b] This field is as defined in Table 6-1.

[c] This field is as defined in Table 6-2.

[d] This field is as defined in Table 6-3.

[e] A frame may contain multiple instances of this field.

[f] There can be multiple instances of the IPv6 extension headers in a single frame. However, they are not ordered in an IPv6 frame as are ordered, e.g., multiple VLAN tags. The instance number for this field is not the usual 0..N−1th instance of an instanced field, but is instead the Next Header value for that header type assigned by the Internet Assigned Numbers Authority.

[g] Since IPv4 and IPv6 headers have similar semantics and since a single frame can be of only IPv4 or IPv6 type but not both, for these frame types, some field codes are reused for the IP equivalents, e.g., protocol types or priority fields. Rule sets that need to treat the same field differently based on IP version are expected to use the ETYPE_LEN field to distinguish IPv4 from IPv6.

[h] Valid LLID values are defined in IEEE Std 802.3, ~~clause~~ 76.2.6.1.3.2.

Sub-attribute *aRuleSetConfig.sClause.sFieldInstance*:

Syntax: Unsigned integer
Range: 0x00 to 0xFF
Remote access: Read/Write
Default value: 0x00
Description: This sub-attribute indicates the instance of the given field within the frame header that is used for matching by this instance of *sClause* sub-attribute. Some fields, such as VLAN tags, may occur in multiple instances in some frames. To distinguish two such fields, the *sFieldInstance* sub-attribute is used in conjunction with the *sFieldCode* sub-attribute. Instances of such fields are numbered starting from 0 in the order in which they are transmitted in the frame. Therefore, for example, C-VLAN tag 0 would be the outermost tag in a frame, immediately after the MAC addresses. In the case of a frame with two C-VLAN tags, C-VLAN tag 1 is the inner tag, closer to the payload of the frame.

Sub-attribute *aRuleSetConfig.sClause.sMaskMsb*:

Syntax: Unsigned integer
Range: 0x00 to 0xFF
Remote access: Read/Write
Default value: 0x00
Description: This sub-attribute indicates the number of bits to ignore on the most significant side of the frame field identified by the *sFieldCode* sub-attribute. The most-significant-bit and least-significant-bit masks (*sMaskMsb* and *sMaskLsb*) are used to reduce the number of field codes and provide flexibility for frame processing rules. A VLAN tag, for instance, is coded as one field (*sFieldCode*). Typically, the processing rules might be using just one of the subfields, e.g., a TPID, CoS, or VID portion of this field. A rule can compare these subfields by using the MSB and LSB masks to isolate the subfield of interest within a larger field.

Sub-attribute *aRuleSetConfig.sClause.sMaskLsb*:

Syntax: Unsigned integer
Range: 0x00 to 0xFF
Remote access: Read/Write
Default value: 0x00
Description: This sub-attribute indicates the number of bits to ignore on the least significant side of the frame field identified by the *sFieldCode* sub-attribute. See additional explanation in the description of the s*MaskMsb* sub-attribute.

Sub-attribute *aRuleSetConfig.sClause.sOperator*:

Syntax: Enumeration
Remote access: Read/Write
Description: This sub-attribute indicates the binary operator for this instance of *aRuleSetConfig.sClause* sub-attribute. The following values are defined:

`NEVER`:	condition never matches.
`EQUAL`:	condition matches if the field is equal to value.
`DIFFERENT`:	condition matches if the field is not equal to value.
`LESS_EQUAL`:	condition matches if the field is less than or equal to value.
`MORE_EQUAL`:	condition matches if the field is greater than or equal to value.
`EXISTS`:	condition matches if the field exists (field value is ignored).
`NOT_EXISTS`:	condition matches if the field does not exist.
`ALWAYS`:	condition always matches.

Sub-attribute *aRuleSetConfig.sClause.sMatchVal*:

Syntax: Unsigned Integer

Size (octets): 120 (max)
Remote access: Read/Write
Description: This sub-attribute represents the numeric value being matched by this instance of *sClause* sub-attribute.

14.4.3.6.1.2 *aRuleSetConfig.sResult* sub-attribute

This sub-attribute represents one of the results of the given frame processing rule, when the given frame matches the combined rule condition. The results of all rules matching a given frame are applied to the frame after all rules have been processed. Multiple results may be applied to each frame. Higher-priority results may overwrite or cancel results of lower-priority rules.

This sub-attribute comprises the following, second-level sub-attributes: *sFrameAction, sQueueId, sFieldCode, sFieldInstance, sMaskMsb, sMaskLsb, sFieldvalue,* and *sCounterIndex*.

Sub-attribute *aRuleSetConfig.sResult.sFrameAction*:

Syntax: Enumeration
Remote access: Read/Write
Description: This sub-attribute indicates the type of result (action on a frame) described by this instance of the *sResult* sub-attribute. Individual values are defined below:

`NOP`: this result has no net effect and does not affect the state of the frame. It may be used as a placeholder result.

`DISCARD`: indicates that all frames matching this rule are to be discarded upon completion of the frame processing operation. This is equivalent to setting the `discard` flag in the frame to `true`.

`FORWARD`: indicates that all frames matching this rule are to be forwarded (not discarded) upon completion of the frame processing operation. This result also sets the `discard` flag in the frame to `false`.

`QUEUE`: indicates the destination queue for frames matching this rule. The destination queue is identified by *sQueueId* sub-attribute.

`SET`: indicates that a specific value is to be written into the selected field in all frames matching this rule. The Field Code, Field Instance, MSB Mask, LSB Mask, and new Field Value are provided in the *sFieldCode, sFieldInstance, sMaskMsb, SMaskLsb,* and *sFieldValue* sub-attributes, respectively. This action does not insert a new field into the frame.

`COPY`: indicates that the value of a selected field (source field) is to be copied into another field (target field). The source field is the field used in the last clause of the rule condition. The target field is identified by *sFieldCode* and *sFieldInstance* sub-attributes. Typically this result is used to copy priority fields, such as IP TOS to IEEE 802.1Q CoS bits, or to copy an inner VLAN tag to an outer one.

`DELETE`: indicates that a field is to be deleted from the processed frame. The field is deleted only when all rules have been processed and no matching higher-priority rule had the `CLEAR_DELETE` result.. The Field Code and Field Instance are provided in the *sFieldCode* and *sFieldInstance* sub-attributes, respectively. This result is commonly used to remove VLAN tags or other encapsulation from a frame.

INSERT: indicates that a field is to be inserted into the processed frame. The field is inserted only when all rules have been processed and no matching higher-priority rule had the CLEAR_INSERT result. The new field is filled with zeros by default. To set this field to a specific value, an additional SET result is provisioned. The Field Code and Field Instance are provided in the *sFieldCode* and *sFieldInstance* sub-attributes, respectively. This result is commonly used to add VLAN tags or other encapsulation to a frame.

REPLACE: represents the combination of INSERT and DELETE results in a single operation. Effectively, the selected field in the frame is replaced with another field. The Field Code and Field Instance are provided in the *sFieldCode* and *sFieldInstance* sub-attributes, respectively. This result is commonly used to translate priority values or VLAN tag values.

CLEAR_DELETE: reverses the decision of a lower-precedence rule to delete the given field in the processed frame. The Field Code and Field Instance are provided in the *sFieldCode* and *sFieldInstance* sub-attributes, respectively.

CLEAR_INSERT: reverses the decision of a lower-precedence rule to insert the given field. The Field Code and Field Instance are provided in the *sFieldCode* and *sFieldInstance* sub-attributes, respectively.

INC_COUNTER: increments programmable counter for frames that match this rule and for octets in those frames.

Sub-attribute *aRuleSetConfig.sResult.sQueueId*:

Syntax: {object type, object instance, queue number} tuple as defined in 14.4.1.1.2.5

Remote access: Read/Write

Description: Object type is equal 0x00-02 or 0x00-03 since only LLIDs and UNI ports have associated queues (see 14.4.1.1.1). This sub-attribute is used only when *sFrameActio*n is set to the value QUEUE.

Sub-attribute *aRuleSetConfig.sResult.sFieldCode*:

See definition of *aRuleSetConfig.sClause.sFieldCode* sub-attribute in 14.4.3.6.1.1.

Description: This sub-attribute represents the code of the field acted upon by the given rule result. This sub-attribute is used when *sFrameAction* is set to one of the following values: SET, COPY, DELETE, INSERT, REPLACE, CLEAR_DELETE, or CLEAR_INSERT.

Sub-attribute *aRuleSetConfig.sResult.sFieldInstance*:

See definition of *aRuleSetConfig.sClause.sFieldInstance* sub-attribute in 14.4.3.6.1.1.

Description: This sub-attribute represents the instance of the field acted upon by the given rule result. This sub-attribute is used when *sFrameAction* is set to one of the following values: SET, COPY, DELETE, INSERT, REPLACE, CLEAR_DELETE, or CLEAR_INSERT.

Sub-attribute *aRuleSetConfig.sResult.sMaskMsb*:

See definition of *aRuleSetConfig.sClause.sMaskMsb* sub-attribute in 14.4.3.6.1.1.

Description: This sub-attribute represents the number of most significant bits of the field that are to be excluded from the action taken by this rule result. This sub-attribute is used only when *sFrameAction* is set to the values SET or COPY.

Sub-attribute *aRuleSetConfig.sResult.sMaskLsb*:

See definition of *aRuleSetConfig.sClause.sMaskLsb* sub-attribute in 14.4.3.6.1.1.

Description: This sub-attribute represents the number of least-significant bits of the field that are to be excluded from the action taken by this rule result. This sub-attribute is used only when *sFrameAction* is set to the values `SET` or `COPY`.

Sub-attribute *aRuleSetConfig.sResult.sFieldValue*:

Syntax: Unsigned integer
Size (octets): 118 (max)
Remote access: Read/Write
Description: This sub-attribute indicates the new value to be written into the field identified by the *sFieldCode* and *sFieldInstance* sub-attributes. This sub-attribute is used only when *sFrameAction* is set to the value `SET`. Values for fields that are not an integral multiple of eight-bit units are right justified and are padded with zeros on the left (most significant) bits.

Sub-attribute *aRuleSetConfig.sResult.sCounterIndex*:

Syntax: Unsigned integer
Size (octets): 0x00-00 to 0x7F-FF
Remote access: Read/Write
Description: This sub-attribute represents the index of the programmable frame counter to be used in a given result. This sub-attribute is used only when *sFrameAction* is set to the value `INC_COUNTER`. The programmable counters are defined in 14.4.6.

14.4.3.6.1.3 *Port Ingress Rule* TLV

A single rule is represented in an eOAMPDU as a series of at least one *Port Ingress Rule* TLV. Each rule can be of an arbitrary complexity and can require more than 128 octets to be fully described, hence exceeding the capacity of a single Variable Container TLV.

The *aRuleSetConfig* attribute is associated with the ONU, PON Port, LLID, UNI Port, or Queue object (see 14.4.1.1). The Variable Container TLV for the *aRuleSetConfig* attribute shall be as specified in Table 14-219.

Table 14-1—*Port Ingress Rule* TLV (0xD7/0x05-01)

Size (octets)	Field (name)	Value	Notes
1	Branch	0xD7	Branch identifier
2	Leaf	0x05-01	Leaf identifier
1	Length	Varies	The size of TLV fields following the `Length` field
1	HeaderIndicator	0x01	Start-of-Rule indicator. For rules that require multiple TLVs, this field may not be present in a given rule TLV.
1	Precedence	Varies	Value of *sPrecedence* sub-attribute. This field is present only when the `HeaderIndicator` is present.
Varies	Clause[0]	Varies	Value of *sClause[0]* sub-attribute (see Table 14-220)
…	…	…	…
Varies	Clause[N−1]	Varies	Value of *sClause[N−1]* sub-attribute (see Table 14-220)
Varies	Result[0]	Varies	Value of *sResult[0]* sub-attribute (see Table 14-221 through Table 14-226)
…	…	…	…

Size (octets)	Field (name)	Value	Notes
Varies	Result[M−1]	Varies	Value of *sResult[M−1]* sub-attribute (see Table 14-221 through Table 14-226)
1	TerminatorIndicator	0x00	End-of-Rule indicator. For rules that require multiple TLVs, this field may not be present in a given rule TLV.

When carried in a Variable Container TLV, the *sClause* sub-attribute shall have the structure as defined in Table 14-220.

Table 14-2—Field structure of *sClause* sub-attribute

Size (octets)	Field (name)	Value	Notes
1	ClauseIndicator	0x02	The value that indicates the beginning of a new clause.
1	FieldCode	Varies	Value of *sFieldCode* sub-attribute, encoded as follows: 0x00: LINK_INDEX field 0x01: DA field 0x02: SA field 0x03: ETYPE_LEN field 0x04: B_DA field 0x05: B_SA field 0x06: I_TAG field 0x07: S_TAG field 0x08: C_TAG field 0x09: MPLS_LSE field 0x0A: IP_TOS_TC field 0x0B: IP_TTL_HL field 0x0C: IP_PT field 0x0D: IPv4_DA field 0x0E: IPv6_DA field 0x0F: IPv4_SA field 0x10: IPv6_SA field 0x11: IPv6_NEXT_HEADER field 0x12: IPv6_FLOWLABEL field 0x13: TCP_UDP_SP field 0x14: TCP_UDP_DP field 0x15: B_TAG field 0x16 to 0x17: reserved 0x18: CUST_0 field 0x19: CUST_1 field 0x1A: CUST_2 field 0x1B: CUST_3 field 0x1C: CUST_4 field 0x1D: CUST_5 field 0x1E: CUST_6 field 0x1F: CUST_7 field 0x20: reserved 0x21: LLID_VALUE field For definitions of individual fields, see Table 6-1.
1	FieldInstance	Varies	Value of *sFieldInstance* sub-attribute
1	MaskMsb	Varies	Value of *sMaskMsb* sub-attribute
1	MaskLsb	Varies	Value of *sMaskLsb* sub-attribute

Size (octets)	Field (name)	Value	Notes
1	Operator	Varies	Value of *sOperator* sub-attribute, encoded as follows: 0x00: NEVER operator 0x01: EQUAL operator 0x02: DIFFERENT operator 0x03: LESS_EQUAL operator 0x04: MORE_EQUAL operator 0x05: EXISTS operator 0x06: NOT_EXISTS operator 0x07: ALWAYS operator
1	MatchValLength	Varies	Length of the MatchVal field. If the Operator field is equal to NEVER, EXISTS, NOT_EXISTS, or ALWAYS, MatchValLength may be equal to 0x00, in which case the MatchVal field is not present.
Varies	MatchVal	Varies	Value of *sMaskVal* sub-attribute.

When carried in a Variable Container TLV, the *sResult* sub-attribute for the frame actions NOP, DISCARD, and FORWARD shall have the structure as defined in Table 14-221.

Table 14-3—Field structure of *sResult* sub-attribute (NOP, DISCARD, and FORWARD actions)

Size (octets)	Field (name)	Value	Notes
1	ResultIndicator	0x03	The value that indicates the beginning of a new result
1	FrameAction	Varies	Value of *sFrameAction* sub-attribute, encoded as follows: 0x00: NOP operation 0x01: DISCARD operation 0x02: FORWARD operation

When carried in a Variable Container TLV, the *sResult* sub-attribute for the frame action QUEUE shall have the structure as defined in Table 14-222.

Table 14-4—Field structure of *sResult* sub-attribute (QUEUE action)

Size (octets)	Field (name)	Value	Notes
1	ResultIndicator	0x03	The value that indicates the beginning of a new result
1	FrameAction	0x03	Value of *sFrameAction* sub-attribute indicating QUEUE operation
4	ObjectType	Varies	Value of *sQueueId* sub-attribute

When carried in a Variable Container TLV, the *sResult* sub-attribute for the frame action SET shall have the structure as defined in Table 14-223.

Table 14-5—Field structure of *sResult* sub-attribute (SET action)

Size (octets)	Field (name)	Value	Notes
1	ResultIndicator	0x03	The value that indicates the beginning of a new result
1	FrameAction	0x04	Value of *sFrameAction* sub-attribute indicating SET operation
1	FieldCode	Varies	Value of *sFieldCode* sub-attribute, encoded as shown in FieldCode field in Table 14-220
1	FieldInstance	Varies	Value of *sFieldInstance* sub-attribute
1	MaskMsb	Varies	Value of *sMaskMsb* sub-attribute
1	MaskLsb	Varies	Value of *sMaskLsb* sub-attribute
1	FieldValueLength	Varies	Length of the FieldValue field
Varies	FieldValue	Varies	Value of *sFieldValue* sub-attribute

When carried in a Variable Container TLV, the *sResult* sub-attribute for the frame action COPY shall have the structure as defined in Table 14-224.

Table 14-6—Field structure of *sResult* sub-attribute (COPY action)

Size (octets)	Field (name)	Value	Notes
1	ResultIndicator	0x03	The value that indicates the beginning of a new result
1	FrameAction	0x05	Value of *sFrameAction* sub-attribute indicating COPY operation
1	FieldCode	Varies	Value of *sFieldCode* sub-attribute, encoded as shown in FieldCode field in Table 14-220
1	FieldInstance	Varies	Value of *sFieldInstance* sub-attribute
1	MaskMsb	Varies	Value of *sMaskMsb* sub-attribute
1	MaskLsb	Varies	Value of *sMaskLsb* sub-attribute

When carried in a Variable Container TLV, the *sResult* sub-attribute for the frame actions DELETE, INSERT, REPLACE, CLEAR_DELETE, and CLEAR_INSERT shall have the structure as defined in Table 14-225.

Table 14-7—Field structure of *sResult* sub-attribute (DELETE, INSERT, REPLACE, CLEAR_DELETE, and CLEAR_INSERT actions)

Size (octets)	Field (name)	Value	Notes
1	ResultIndicator	0x03	The value that indicates the beginning of a new result
1	FrameAction	Varies	Value of *sFrameAction* sub-attribute, encoded as follows: 0x06: DELETE operation 0x07: INSERT operation 0x08: REPLACE operation 0x09: CLEAR_DELETE operation 0x0A: CLEAR_INSERT operation

Size (octets)	Field (name)	Value	Notes
1	FieldCode	Varies	Value of *sFieldCode* sub-attribute, encoded as shown in `FieldCode` field in Table 14-220
1	FieldInstance	Varies	Value of *sFieldInstance* sub-attribute

When carried in a Variable Container TLV, the *sResult* sub-attribute for the frame action `INC_COUNTER` shall have the structure as defined in Table 14-226.

Table 14-8—Field structure of *sResult* sub-attribute (`INC_COUNTER` action)

Size (octets)	Field (name)	Value	Notes
1	ResultIndicator	0x03	The value that indicates the beginning of a new result
1	FrameAction	0x0B	Value of *sFrameAction* sub-attribute indicating `INC_COUNTER` operation
2	CounterIndex	Varies	Value of *sCounterIndex* sub-attribute