12. Operations, administration and maintenance (OAM)

12.1 Overview

This clause provides operations, administration and maintenance (OAM) functions supported by RPR stations. The methods defined are intended to be scalable, to cause insignificant overhead for ring traffic, and to cause insignificant overhead on software and hardware devices. The protocol resides in the MAC control sublayer, as shown in the shaded OAM region of Figure 12.1.

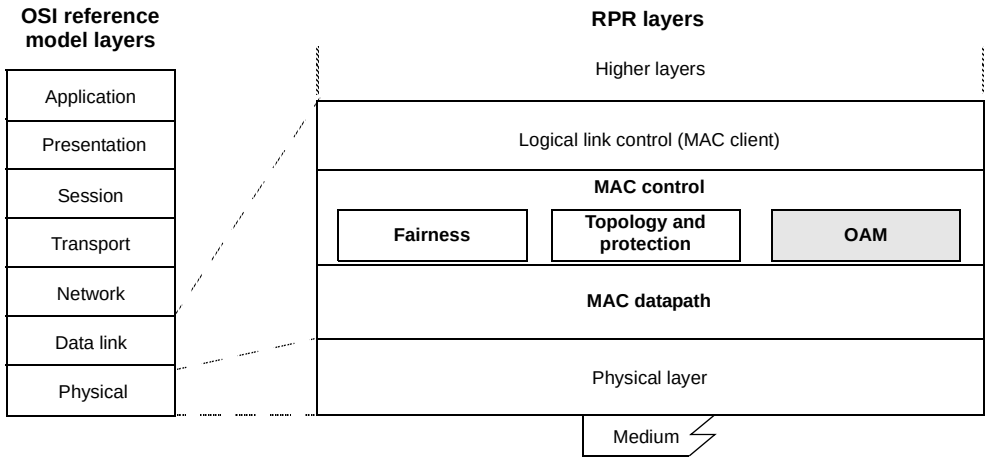

Figure 12.1—OAM control entity relationship to the ISO/IEC OSI reference model

The services and features provided are as follows:

a) Determine/validate connectivity between any two stations on the ring.
b) Determine/validate transit path operation for any service class.
c) Operate without relying on a master station.
d) Provide a mechanism to help in misorder prevention.

It is not within the scope of the OAM functions to detect intermittent failures.

12.1.1 Protocol overview

Management functional areas pertinent to RPR are as follows:

a) configuration management
b) fault management
c) performance management

Configuration management exercises control over, identifies, collects data from, and provides data to stations and the connections between stations. Configuration management is responsible for the installation of stations, their interconnection into a network, and provisioning.

Fault (or maintenance) management enables the detection, isolation, and correction of abnormal operation of the stations and the network. It is responsible for detecting and processing any faults, as well as to report them to the management system.

Performance management evaluates and reports upon the behavior and effectiveness of stations and the network. Performance management provides mechanisms to measure service quality, by monitoring the system performance. It also reports statistics information to the management system.

In order to improve fault and performance management capability, e.g., to allow fault detection, some in-band OAM functions are provided.

OAM functions in a network are performed on hierarchical levels. Different types of interfaces support different levels of OAM functionality. A physical layer based on SONET/SDH includes extensive OAM functions, while a physical layer based on PacketPHYs has a lower level of OAM support.

This standard re-uses existing in-band OAM mechanisms provided by the PHY layers. In-band OAM functionality used in upper layers is outside the scope of this standard.

12.1.2 OAM functions supported by RPR

The OAM functions are based on special frames, sent between stations on a ring, that test the operational status of a path between stations.

The OAM frame types supported are as follows:

a) Echo—For on demand connectivity monitoring and fault localization on the path between stations, or for client determination of LRTT.
b) Flush—For client controlled misorder prevention when changing preferred ring direction of a given flow, or for client determination of RRTT.
c) Organization specific—For encoding organization specific OAM information.

All stations support echo request/response frames and flush frames.

12.1.3 Fault management

Fault management includes alarm surveillance, fault localization, fault correction, and testing. Alarm surveillance provides the capability to monitor failures detected in stations. In support of alarm surveillance, stations perform checks on hardware and software in order to detect failures, and generate alarms for such failures.

Fault localization determines the root cause of a failure. In addition to the initial failure information, it can use failure information from other entities in order to correlate and localize the fault.

Fault correction is responsible for the repair of a fault and for the control of procedures that use redundant resources to replace equipment or facilities that have failed. In the cases of fiber cuts or station failures, protection switching restores service.

Testing performs repair functions using testing and diagnostic routines. Testing is characterized as the application of signals/messages and their measurement. Echo request/response is one example of a testing routine.

The fault management mechanisms are useful to check the reachability at the MAC layer between two stations on the ring, especially when there are some failures (e.g., one station on the ring steals frames addressed to other stations) that are not detected at layer 1.

This clause defines one fault management mechanism. It is an on-demand in-service echo request/response mechanism (see 12.1.4) used for troubleshooting the RPR network (reactive mechanism).

In addition, Annex L describes a method for using echo frames for fault detection (proactive mechanism) by continuous connectivity verification between any pair of stations.

12.1.4 Echo operations

This standard allows the management system to request an echo request/response operation to a specified destination in order to check the reachability of a station. An echo operation allows a frame to be inserted at one station in the ring, and an echo response returned by another station through the same or opposite ringlet, as illustrated in Figure 12.2. Service primitive parameters allow the client to specify the absolute, reverse, or default ringlet for the response (see Table 12.2).

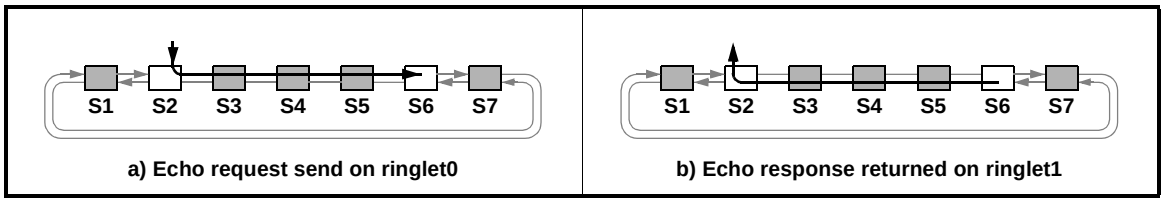

Figure 12.2—Echo request/response operations

Echo request/response frames can be assigned to any service class, and contain user specified bytes up to the maximum permitted frame size. The echo request *userData* is copied into the response frame.

12.1.5 Flush operations

A flush has the effect of clearing the selected ringlet of previously sourced traffic. A flush is expected to be used when changing the ringlet selection algorithm, when revised ringlet selection protocols are necessary to access all stations (for steering protection), or to improve bandwidth utilization (for wrap protection).

A flush is a special control frame that is sent from a station to itself, as illustrated in Figure 12.3-a. Although distinct flushes (one for ringlet0 and one for ringlet1) are necessary to ensure complete delivery of all previously sourced traffic, one flush is often sufficient to flush the relevant traffic. The flush frame can also flush all previously sourced traffic from a wrapped ring, as illustrated in Figure 12.3-b.

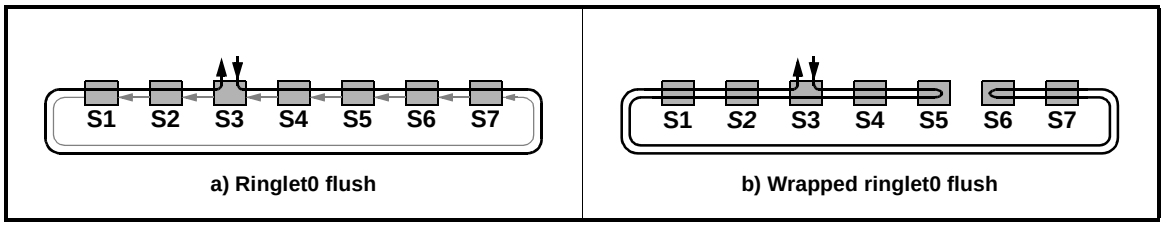

Figure 12.3—Flushing on an unprotected and wrapped rings

The flush frames can be assigned to any service class, with distinct flush action results, as follows:

a) ClassA. Previously-sourced primary transit queue (PTQ) traffic is flushed.
(For a single transit queue station, classB and classC traffic are also flushed).

b) ClassB or classC. Previously-sourced PTQ and secondary transit queue (STQ) traffic is flushed.
Since the classA traffic has precedence, others' PTQs are also flushed.

Flushing is not possible on a steering protection ring, since traffic is discarded (rather than wrapped) at the endpoints. However, an endpoint-addressed echo can be used in a similar fashion. Annex L illustrates the use of echo frames for flushing transit queues.

12.1.6 Organization specific operations

An RPR station may use the optional organization specific OAM capability to implement additional OAM functions not specified by this standard.

If implemented, the organization specific MAC client service request generates and transmits an organization specific frame. Similarly, the receipt of an organization specific frame generates a MAC client service indication.

12.2 Terminology and variables

12.2.1 Common state machine definitions

The following state-machine definitions are used multiple times within this clause:

RR_DEFAULT
RR_REVERSE
RR_RINGLET0
RR_RINGLET1
: The enumerated *responseRinglet* values (see 12.3.1.3).

12.2.2 Common variables

The following state machine inputs are used multiple times within this clause.

sesThreshold
: Configured. Number of *scffErrors* (see 12.5.1.2) to declare a second as a severely errored second. Table 12.1 shows the *sesThreshold* values for the PHYs described in this standard.
: Range: [1, 512], in integer increments
: Default: the next higher integer of 0.000001 * (*lineRate* * *advertisementRatio*)

Table 12.1—*sesThreshold* example values

PHY	*sesThreshold*
1 Gb/s PacketPHY	2
OC-48	3
OC-192	13
10 Gb/s PacketPHY	

12.2.3 Common routines

There are no common routines defined within this clause.

12.2.4 Literals and routines defined in other clauses

This clause references the following literals and variables defined in Clause 6:

mac_protection
OAM_ECHO_REQ
OAM_FLUSH_REQ
OAM_ORG_REQ
    Control request *opcode* values.
Q_OAM_REQ
    Queue for OAM request frames (TBD).
RI_0
RI_1
RI_DEFAULT
ringlet_id

This clause references the following literals and routines defined in Clause 7:

Dequeue(*queue*)
Enqueue(*queue, frame*)
lineRate
myMacAddress
Q_RX_ECHO_REQ
Q_RX_ECHO_RSP
Q_RX_FLUSH
Q_RX_ORG
Q_TX_RS

This clause references the following literals defined in Clause 9:

CT_OAM_ECHO_RSP

This clause references the following literals and routines defined in Clause 10:

advertisementRatio

12.3 OAM frame formats

OAM functions use frames in the control frame format (see 9.3) with *controlType* set to one of the assigned OAM codes. Like most other frames, OAM frames are not transmitted over failed spans, even if the span failure is unidirectional.

12.3.1 Echo request/response payload

Figure 12.4 illustrates the echo request/response frame payload, with the *controlDataUnit* portion of the frame indicated.

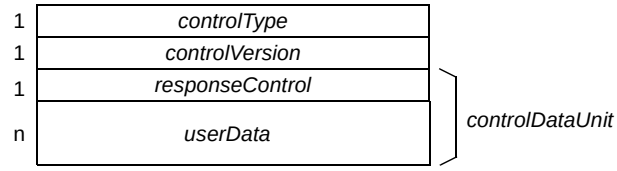

Figure 12.4—Echo request/response frame payload format

The *responseControl* field is shown in Figure 12.5. The *responseControl* field carries the same values in the response as were provided in the request, although no processing is done with them on receipt of the response.

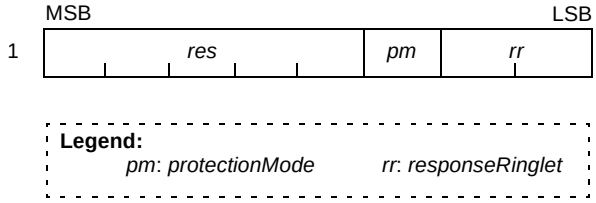

Figure 12.5—*responseControl* field format

12.3.1.1 *res*: A 5-bit reserved field.

12.3.1.2 *protectionMode*: A bit that indicates to the echo request receiving station whether to protect the echo response frame in case of a ring failure. A value of 0 indicates that the echo response is not protected, while a value of 1 indicates that the echo response is protected by the MAC. It uses the protection configuration method of the receiving station, and also sets mac_protection to this value if the configured protection method is wrapping.

12.3.1.3 *responseRinglet*: The 2-bit *responseRinglet* field is interpreted by the receiving station to decide through which ringlet the echo response frame will be transmitted back to the source station. The *responseRinglet* field is encoded as shown in Table 12.2.

Table 12.2—*responseRinglet* values

Value	Name	Description
0	RR_RINGLET0	Reply on ringlet0
1	RR_RINGLET1	Reply on ringlet1
2	RR_REVERSE	Reply on reverse ringlet
3	RR_DEFAULT	Reply on default ringlet

The RR_DEFAULT option allows the echo responder to determine the echo response frame path, by setting ringlet_id to RR_DEFAULT before invoking ringlet selection. The RR_RINGLET0, RR_RINGLET1, and RR_REVERSE options allow the echo request source station to specify the echo response path.

12.3.1.4 *userData*: A n-byte optional field whose content is determined by the client. When an echo request is received, the receiving station copies the *userData* field from the echo request frame to a generated echo response frame.

12.3.2 Flush frame

Figure 12.6 illustrates the flush frame payload with the *controlDataUnit* portion of the frame indicated.

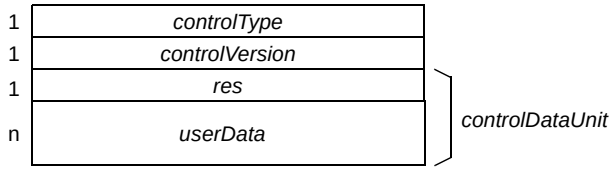

Figure 12.6—Flush frame payload format

12.3.2.1 *res*: A reserved byte.

12.3.2.2 *userData*: An n-byte optional field whose content is determined by the client.

12.3.3 Organization specific frame

Figure 12.7 illustrates the organization specific frame payload, with the *controlDataUnit* portion of the frame indicated.

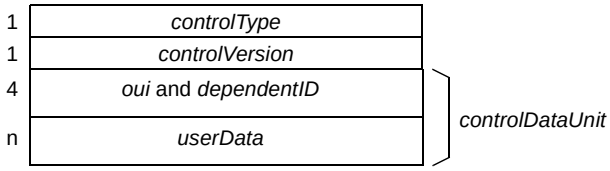

Figure 12.7—Organization specific frame payload format

The organization specific OAM frame encodes the organization specific OAM information of the station, and is optional. If a destination station does not support the organization specific OAM frame it is allowed to discard the frame without triggering any alarm or indication.

The *controlDataUnit* field for organization specific frames is illustrated in Figure 12.8.

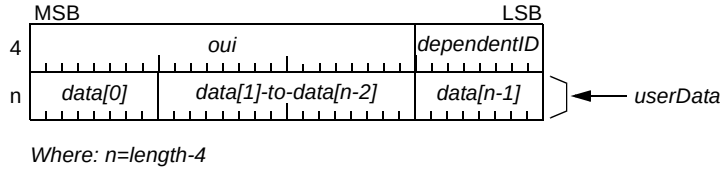

Figure 12.8—Organization specific *controlDataUnit* field format

12.3.3.1 *oui*: A 24-bit organizationally unique identifier (OUI) field supplied by the IEEE/RAC for the purpose of identifying the organization supplying the (unique within the organization, for this specific context) 8-bit *dependentID*.

12.3.3.2 *dependentID*: An 8-bit field supplied by the *oui*-specified organization. The concatenation of the *oui* and *dependentID* provide a unique (within this context) identifier for each distinct form of the following *userData*.

12.3.3.3 *userData*: An *n*-byte optional field whose content is implied by the preceding *oui/dependentID* value. The content in this field is beyond the scope of this standard.

To reduce the likelihood of error, the mapping of OUI values to the *oui/dependentID* fields are illustrated in Figure 12.9. For the purposes of illustration, specific OUI and *dependentID* example values have been assumed.

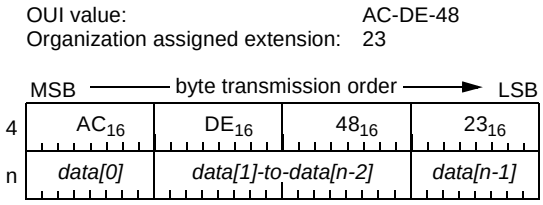

Figure 12.9—Mapping of organization specific values

12.4 OAM service primitives

12.4.1 Echo MA_CONTROL.request

The client uses the MA_CONTROL.request function (see 6.4.3) to request the generation of an echo request frame. The semantics for the MA_CONTROL.request of an echo request frame are as follows:

```
MA_CONTROL.request
(
    OAM_ECHO_REQ,
    destination_address,
    service_class,
    ringlet_id,          // optional
    mac_protection,      // optional
    response_ringlet,    // optional
    user_data            // optional
)
```

The parameters of the MA_CONTROL.request of an echo request frame are described below.

OAM_ECHO_REQ
: Opcode that indicates an echo request frame generation operation request.

destination_address
: Specifies an individual MAC address of any station in the ring including the local station.
: Sets the *da* (destination MAC address) of the transmitted echo request frame (see 12.3.1).

service_class
: The class of service requested by the MAC client (see Table 6.1).
: The MAC entity uses this value to
: a) Select the value of the *sc* field (see 9.6.4) of the transmitted echo request frame.
: b) Specify the MAC treatment of the transmitted frame (see 7.9.1).

ringlet_id
: Indicates the ringlet choice of the client (see Table 6.3 and 7.8).
: This value selects the *ri* (ringlet identifier) field value of the transmitted echo request frame (before any protection based change) (see 9.6.1). If the *ringlet_id* parameter is omitted, the MAC uses its default algorithm to determine which ringlet to use.

mac_protection
: Indicates a choice of whether the MAC provides protection for the echo request frame (see 7.8). The mac_protection field also sets the value of the *protectionMode* bit in the *responseControl* field (see 12.3.1.2).
: TRUE—The MAC provides protection and *protectionMode* is set to 1.
: FALSE—The MAC does not provide protection and *protectionMode* is cleared to 0.
: (null)—The default value is TRUE.

response_ringlet
: Sets the value of the *responseRinglet* field in the *responseControl* field (see Table 12.2 and 12.3.1.3).
: (null)—The default value is RR_DEFAULT.

user_data
: If present, user_data sets the value of the *userData* field in the echo request frame (see 12.3.1.4).

12.4.2 Echo MA_CONTROL.indication

The MAC uses the MA_CONTROL.indication function (see 5.4.4) to indicate to the client the reception of an echo response frame. The semantics of the MA_CONTROL.indication of an echo response frame are as follows:

```
MA_CONTROL.indication
(
    OAM_ECHO_IND,
    source_address,
    service_class,
    ringlet_id,
    protection_mode,
    response_ringlet,
    user_data
)
```

The parameters of the MA_CONTROL.indication of an echo response frame are described below.

OAM_ECHO_IND
: An *opcode* value that indicates an echo response frame arrival to the MAC sublayer.

source_address
: The value of the *sa* field in the incoming echo response frame, as specified in 9.3.2.4.

service_class
: The service class at which the echo response frame was sent, as specified in Table 6.1 (see also Table 7.1 of 7.1.1).

ringlet_id
: The setting of the *ri* bit in the echo response frame header, as specified in Table 6.2 (see also 9.6.1).

protection_mode
: The setting of the *protectionMode* bit in the *responseControl* field of the echo response frame (see 12.3.1.2).

response_ringlet
: The setting of the *responseRinglet* field in the *responseControl* field of the echo response frame (see 12.3.1.3).

user_data
: Data from the *userData* field of the incoming echo response frame (see 12.3.1.4).

12.4.3 Flush MA_CONTROL.request

The client uses the MA_CONTROL.request function (see 6.4.3) to request the generation of a flush frame. The semantics for the MA_CONTROL.request of a flush frame are as follows:

```
MA_CONTROL.request
(
    OAM_FLUSH_REQ,
    service_class,
    ringlet_id,          // optional
    mac_protection,      // optional
    user_data            // optional
)
```

The parameters of the MA_CONTROL.request of a flush frame are described below.

OAM_FLUSH_REQ
Opcode that indicates a flush frame generation operation request.
service_class
Indicates the class of service requested by the MAC client, as specified in Table 6.1.
The service_class value selects the value of the *sc* field (see 9.6.4) of the transmitted flush frame, and to indicate the requested MAC treatment of the transmitted frame (see 7.9.1).
ringlet_id
Indicates the ringlet choice of the client, as specified in Table 6.2.
The ringlet_id value selects the value of the *ri* (ringlet identifier) field (see 9.6.1) of the transmitted flush frame, before any protection based change (see 7.8). If the ringlet_id parameter is omitted, the MAC uses its default algorithm to determine which ringlet to use.
mac_protection
Indicates a choice of whether the MAC provides protection for the flush frame (see 7.8).
TRUE—The MAC shall provide protection for the flush frame.
FALSE—The MAC shall not provide protection for the flush frame.
(null)—The default value is TRUE.
user_data
Data for the *userData* field of the flush frame (see 12.3.2.2).

12.4.4 Flush MA_CONTROL.indication

The MAC uses the MA_CONTROL.indication function (see 5.4.4) to indicate to the client the reception of a flush frame. The semantics of the MA_CONTROL.indication of an flush frame are as follows:

```
MA_CONTROL.indication
(
    OAM_FLUSH_IND,
    service_class,
    ringlet_id,
    user_data
)
```

The parameters of the MA_CONTROL.indication of a flush frame are described below.

OAM_FLUSH_IND
An *opcode* value that indicates a flush frame arrival to the MAC sublayer.
service_class
Indicates the service class at which the flush frame was sent, as specified in Table 6.1 (see also Table 7.2 of 7.1.1).
ringlet_id
Indicates the setting of the *ri* bit (see 9.6.1) in the flush frame header, as specified in Table 6.2.
user_data
Data from the *userData* field of the incoming flush frame, as specified in 12.3.2.2.

12.4.5 Organization specific MA_CONTROL.request

The client uses the MA_CONTROL.request function (see 6.4.3) to request the generation of an organization specific frame. The semantics for the MA_CONTROL.request of a organization specific frame are as follows:

```
MA_CONTROL.request
(
    OAM_ORG_REQ,
    destination_address,
    service_class,
    ringlet_id,          // optional
    mac_protection,      // optional
    organization,
    user_data            // optional
)
```

The parameters of the MA_CONTROL.request of a organization specific frame are described below.

OAM_ORG_REQ
: An *opcode* value that indicates an organization specific frame generation operation request.

destination_address
: Specifies an individual MAC address of any station in the ring including the local station.
: Sets the *da* (destination MAC address) of the transmitted organization specific frame (see 12.3.1).

service_class
: The class of service requested by the MAC client (see Table 6.1).
: The MAC entity uses this value to
    a) Select the *sc* field (see 9.6.4) value in the transmitted organization specific frame.
    b) Specify the MAC treatment of the transmitted frame (see 7.9.1).

ringlet_id
: Indicates the ringlet choice of the client (see Table 6.2 and 7.8).
: This value selects the *ri* (ringlet identifier) field value of the transmitted organization specific frame, before any protection based change (see 9.6.1). If the ringlet_id parameter is omitted, the MAC uses its default algorithm to determine which ringlet to use.

mac_protection
: Indicates whether the MAC provides protection for the organization specific frame (see 7.8).
    TRUE—The MAC shall provide protection for the organization specific frame.
    FALSE—The MAC shall not provide protection for the organization specific frame.
    (null)—The default value is TRUE.

organization
: Sets the value of the *oui* and *dependentID* fields in the organization specific frame (see 12.3.3.1 and 12.3.3.2).

user_data
: If present, sets the value of the *userData* field in the organization specific frame (see 12.3.3.3).

12.4.6 Organization specific MA_CONTROL.indication

The MAC uses the MA_CONTROL.indication function (see 5.4.4) to indicate to the client the reception of an organization specific frame. The semantics of the MA_CONTROL.indication of an organization specific frame are as follows:

```
MA_CONTROL.indication
(
    OAM_ORG_IND,
    source_address,
    service_class,
    ringlet_id,
    organization,
    user_data
)
```

The parameters of the MA_CONTROL.indication of an organization specific frame are described below.

OAM_ORG_IND
: An *opcode* value that indicates an organization specific frame arrival to the MAC sublayer.

source_address
: The *sa* filed in the incoming organization specific frame (see 9.3.2.4).

service_class
: The service class at which the organization specific frame was sent (see Table 6.1 and Table 7.1).

ringlet_id
: The setting of the *ri* bit in the organization specific frame header (see Table 6.2 and 9.6.1).

organization
: Indicates the setting of the *oui* and *dependentID* fields in the organization specific frame (see 12.3.3.1 and 12.3.3.2).

user_data
: Data from the *userData* field of the incoming organization specific frame (see 12.3.3.3).

12.5 OAM state machines

12.5.1 OamFrameTransmit state machine definitions

12.5.1.1 OamFrameTransmit state machine definitions

Q_OAM_REQ
: See 12.2.4.

OAM_ECHO_REQ

OAM_FLUSH_REQ

OAM_ORG_REQ
: See 12.2.4.

12.5.1.2 OamFrameTransmit state machine variables

frame
: A control frame being prepared for transmissions.

12.5.1.3 OamFrameTransmit state machine routines

Dequeue(*queue*)
See 12.2.4.
EchoRequestFrame(*frame*)
Formats the remaining frame field values for an echo request frame (see 12.3.1).
Enqueue(*queue, frame*)
See 12.2.4.
FlushRequestFrame(*frame*)
Formats the remaining frame field values for an flush request frame (see 12.3.2).
OrgRequestFrame(*frame*)
Formats the remaining frame field values for an organization specific request frame (see 12.3.3).

12.5.1.4 OamFrameTransmit state table

The OamFrameTransmit state machine specified in Table 12.3 implements the functions necessary for OAM frame request processing by the source station. In the case of any ambiguity between the text and the state machine, the state machine shall take precedence. The notation used in the state table is described in 3.4.

Table 12.3—OamFrameTransmit state table

Current state		Row	Next state	
state	**condition**		**action**	**state**
START	(frame = Dequeue(Q_OAM_REQ)) != NULL	1	—	FORM
	—	2	—	START
FORM	frame.controlType == OAM_ECHO_REQ	3	frame = EchoRequestFrame(frame);	SEND
	frame.controlType == OAM_FLUSH_REQ	4	frame = FlushRequestFrame(frame);	
	frame.controlType == OAM_ORG_REQ	5	frame = OrgRequestFrame(frame);	
SEND	—	6	Enqueue(Q_TX_RS, frame);	START

Row 12.3-1: A OAM frame generation operation is requested by the MAC client.
Row 12.3-2: Wait for OAM frame generation request.

Row 12.3-3: If the client opcode indicates echo request, generate an echo request frame.
When the MAC services an echo request destined to itself, the echo request is still transmitted onto the ring.
Row 12.3-4: If client opcode indicates flush, generate a flush request frame.
Row 12.3-5: If the client opcode indicates organization specific, generate an organization specific frame.

Row 12.3-6: Forward frame to ringlet selection.

12.5.2 OamFrameReceive state machine

The OamReceiving state machine implements the functions necessary for OAM frame reception, including echo response formation/transmission at the target station.

12.5.2.1 OamFrameReceive state machine definitions

CT_OAM_ECHO_RSP
Q_RX_ECHO_REQ
Q_RX_ECHO_RSP
Q_RX_FLUSH
Q_RX_ORG
Q_TX_RS
    See 12.2.4.
RINGLET_0
RINGLET_1
    See 12.2.4.
RR_RINGLET0
RR_RINGLET1
RR_DEFAULT
    Enumerated values in an echo request, used to specify the echo response ringlet (see Table 12.2).

12.5.2.2 OamFrameReceive state machine variables

mac_protection
myMacAddress
    See 12.2.4.
ringlet_id
    See 12.2.4.
rxFrame
    A received control frame.
txFrame
    A control frame being prepared for transmissions.

12.5.2.3 OamFrameReceive state machine routines

Dequeue(*queue*)
Enqueue(*queue*, *frame*)
    See 12.2.4.
OamEchoIndication( )
    Sends an indication to the client, with a specified MA_CONTROL.indication opcode value of OAM_ECHO_IND and the corresponding operands (see 6.4.4).
OamFlushIndication( )
    Sends an indication to the client, with a specified MA_CONTROL.indication opcode value of OAM_FLUSH_IND and the corresponding operands (see 6.4.4).
OamOrganizationIndication( )
    Sends an indication to the client, with a specified MA_CONTROL.indication opcode value of OAM_ORG_IND and the corresponding operands (see 6.4.4).

12.5.2.4 OamFrameReceive state table

The OamFrameReceive state table is specified in Table 12.4. In the case of any ambiguity between the text and the state machine, the state machine shall take precedence. The notation used in the state table is described in 3.4.

Table 12.4—OamFrameReceive state table

Current state		Row	Next state	
state	**condition**		**action**	**state**
START	(rxFrame = Dequeue(Q_RX_ECHO_REQ)) != NULL	1	—	FIRST
	(rxFrame = Dequeue(Q_RX_ECHO_RSP)) != NULL	2	OamEchoIndication(rxFrame);	START
	(rxFrame = Dequeue(Q_RX_FLUSH)) != NULL	3	OamFlushIndication(rxFrame);	
	(rxFrame = Dequeue(Q_RX_ORG)) != NULL	4	OamOrganizationIndication(rxFrame);	
	—	5	—	
FIRST	rxFrame.responseRinglet==RR_RINGLET0	6	ringlet_id = RI_0;	MODE
	rxFrame.responseRinglet==RR_REVERSE && rxFrame.ri == RINGLET_1	7		
	rxFrame.responseRinglet == RR_RINGLET1	8	ringlet_id = RI_1;	
	rxFrame.responseRinglet==RR_REVERSE && rxFrame.ri == RINGLET_0	9		
	—	10	ringlet_id = RI_DEFAULT;	
MODE	rxFrame.protectionMode != 0	11	mac_protection = TRUE;	FINAL
	—	12	mac_protection = FALSE;	

FINAL	—	13	txFrame = rxFrame;
			txFrame.controlType = CT_OAM_ECHO_RSP;
			txFrame.da = rxFrame.sa;
			txFrame.sa = myMacAddress;
			Enqueue(Q_TX_RS, txFrame);
				START

Row 12.4-1: An echo request frame is received.
Row 12.4-2: An echo response frame sets the MA_CONTROL.indication opcode and operands.
Row 12.4-3: A flush frame sets the MA_CONTROL.indication opcode and operands.
Row 12.4-4: An organization specific frame sets MA_CONTROL.indication opcode and operands.
Row 12.4-5: Wait for the next OAM frame.

Row 12.4-6: If the echo request indicates RR_RINGLET0, place the echo response on ringlet0.
Row 12.4-7: If a ringlet1 echo request indicates RR_REVERSE, place the echo response on ringlet0.
Row 12.4-8: If the echo request indicates RR_RINGLET1, place the echo response on ringlet1.

Row 12.4-9: If a ringlet0 echo request indicates RR_REVERSE, place the echo response on ringlet1.
Row 12.4-10: If an echo request indicates RR_DEFAULT, place the echo response on the default ringlet.

Row 12.4-11: If the echo request indicates protection, select MAC protected mode transmission.
Row 12.4-12: If the echo request indicates no protection, select MAC unprotected mode transmission.

Row 12.4-13: Generate an echo response frame, copy echo request fields, and forward to ringlet selection. When the MAC receives an echo request from itself, the echo response is still transmitted onto the ring.

The expected time between an echo request and the echo response is a function of the ring size, distance to the responding station, processing time of the responding station, and the service class chosen for the echo frame. However, the time elapsed between an echo request frame reception (a frame arrives in the queue used by Row 12.4-1) and the generation of the corresponding echo response (the frame leaves the queue used by Row 12.4-13) shall be no more than 10 milliseconds, when echo requests are received no more frequently than once every 100 milliseconds.

12.5 Performance monitoring

Stations may accumulate performance parameters related to RPR to enable the detection of developing failures before a total outage is detected.

The following subclauses define the RPR performance parameters related to error monitoring and availability. Stations may support some, all, or none of these parameters.

12.5.1 Performance monitoring counters

The performance monitoring counters, described in the following subclauses, are based on the concept of available and unavailable seconds, as adapted from ANSI T1.231-1997 6.5.4.2.8 (see 12.5.2).

12.5.1.1 *unavailableSeconds*[1]

The *unavailableSeconds* counter is incremented onced during each second for which the RPR service is unavailable (see 12.5.2).

12.5.1.2 *scffErrors*

The *scffErrors* counter is incremented during an available second, once for each observed SCFF with either of the following properties:

a) The SCFF's observed *parity* bit differs from its locally computed value.
b) The SCFF's observed *fcs* field differs from its locally computed value.

12.5.1.3 *erroredSeconds*

The *erroredSeconds* counter is incremented once during an available second in which any of the following occurs:

a) An SCFF's observed *parity* bit differs from its locally computed value.
b) An SCFF's observed *fcs* field differs from its locally computed value.
c) An RPR alarm (see 12.1.5) was present

[1]Adapted from ANSI T1.231-1997 6.5.4.2.8

d) A PHY failure (see 8.2) was present.

12.5.1.4 *severelyErroredSeconds*

The *severelyErroredSeconds* counter is incremented once during an available second in which any of the following occurs:

a) The value of *scffErrors* increased by at least *sesThreshold* counts.
b) An RPR alarm (see 12.1.5) was present.
c) A PHY failure (see 8.2) was present.

12.5.2 Available and unavailable seconds

The update of the preceding performance monitoring counts depends on the available/unavailable classification of the seconds being measured. This classification is based on the presence or absence of severely errored second (SES) intervals, as illustrated in Figure 12.10.

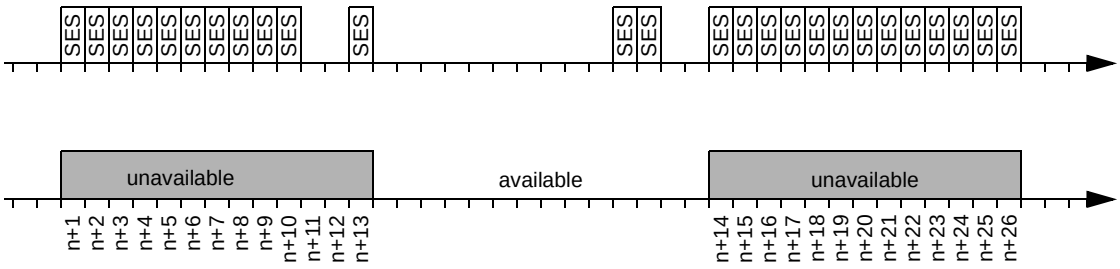

Figure 12.10—Available second classification

Once available, the RPR service transitions to unavailable at the onset of 10 contiguous SESs, where the 10 initial SESs are included in unavailable time. Once unavailable, the RPR service transitions to available at the onset of 10 contiguous seconds, none of which are classified as an SES. The 10 seconds with no SESs are included in the available time.

In Figure 12.10, for example, the value of *n+i* represents the *unavailableSeconds* counter.

A severely errored second (SES) is classified as a second during which any of the following occurs:

a) At least *sesThreshold* SCFFs are in error, in either of the following ways:
   1) The SCFF's observed *parity* bit differs from its locally computed value.
   2) The SCFF's observed *fcs* field differs from its locally computed value.
b) An RPR alarm (see 12.1.5) is present.
c) A PHY failure (see 8.2) is present.

12.6 Protocol Implementation Conformance Statement (PICS) proforma for Clause 12[2]

12.6.1 Introduction

The supplier of a protocol implementation that is claimed to conform to Clause 12, Operations, administration and maintenance (OAM), shall complete the following Protocol Implementation Conformation Statement (PICS) proforma.

A detailed description of the symbols used in the PICS proforma, along with instructions for completing the same, can be found in Annex A of IEEE Std 802.1Q-1998.

12.6.2 Identification

12.6.2.1 Implementation identification

Supplier[a]	
Contact point for enquiries about the PICS[a]	
Implementation Name(s) and Version(s)[a,c]	
Other information necessary for full identification—e.g., name(s) and version(s) for machines and/or operating systems; System Name(s))[b]	
a—Required for all implementations. b—May be completed as appropriate in meeting the requirements for the identification. c—The terms Name and Version should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model).	

12.6.2.2 Protocol summary

Identification of protocol standard	IEEE Std 802.17-200x, Resilient packet ring access method and physical layer specifications, Clause 12, Operations, administration and maintenance (OAM)
Identification of amendments and corrigenda to this PICS proforma that have been completed as part of this PICS	
Have any Exception items been required? No [ ] Yes [ ] (The answer Yes means that the implementation does not conform to IEEE Std 802.17-200x.)	

Date of Statement	

[2]*Copyright release for PICS proformas:* Users of this standard may freely reproduce the PICS proforma in this clause so that it can be used for its intended purpose and may further publish the completed PICS.

12.6.3 PICS tables for Clause 12

12.6.3.1 Major options

Item	Feature	Subclause	Value/Comment	Status	Support
ORG*	Organization specific frames	12.3.3	Adds organization specific functionality.	O	Yes [ ] No [ ]
PM*	Performance monitoring	12.5	accumulate performance parameters	O	Yes [ ] No [ ]

12.6.3.2 State machines

Item	Feature	Subclause	Value/Comment	Status	Support
SM1	OamFrameTransmit state machine	12.5.1	OAM request frames generated in response to a MAC client service request.	M	Yes [ ]
SM2	OamFrameReceive state machine	12.5.2	OAM response frames generate a MAC client service indication; an echo request generates an echo response.	M	Yes [ ]

12.6.3.3 Frame formats

Item	Feature	Subclause	Value/Comment	Status	Support
FF1	Echo frame formats	12.3.1	Echo request/response formats	M	Yes [ ]
FF2	Flush frame format	12.3.2	Flush frame format	M	Yes [ ]
FF3	Organization specific frame format	12.3.2	Organization specific frame format	ORG:M	Yes [ ]

12.6.3.4 Service primitives

Item	Feature	Subclause	Value/Comment	Status	Support
SP1	MA_CONTROL.request with an opcode value of OAM_ECHO_REQ	12.4.1	Echo request/response formats	M	Yes [ ]
SP2	MA_CONTROL.indication with an opcode value of OAM_ECHO_IND	12.4.2	Echo request/response formats	M	Yes [ ]
SP3	MA_CONTROL.request with an opcode value of OAM_ECHO_REQ	12.4.3	Echo flush formats	M	Yes [ ]
SP4	MA_CONTROL.indication with an opcode value of OAM_ECHO_IND	12.4.4	Echo flush formats	M	Yes [ ]
SP5	MA_CONTROL.request with an opcode value of OAM_ORG_REQ	12.4.5	Organization specific formats	ORG:M	Yes [ ]
SP6	MA_CONTROL.indication with an opcode value of OAM_ORG_IND	12.5.2	Organization specific formats	ORG:M	Yes [ ]

12.6.3.5 Performance monitoring

Item	Feature	Subclause	Value/Comment	Status	Support
PM1	*scffErrors*	12.5.1.2	counts errors	PM:O	Yes [ ] No [ ]
PM2	*erroredSeconds*	12.5.1.3	counts seconds	PM:O	Yes [ ] No [ ]
PM3	*severelyErroredSeconds*	12.5.1.4	counts seconds	PM:O	Yes [ ] No [ ]
PM4	*unavailableTime*	12.5.1.1	counts 1 second intervals	PM:O	Yes [ ] No [ ]
PM5	*scffErrors*, *erroredSeconds*, *severelyErroredSeconds*	12.5.1.2 12.5.1.3 12.5.1.4	accumulated during available time only	PM:M	Yes [ ]