

Definitions and Managed Objects for
Operations, Administration, and Maintenance (OAM) Functions on
Ethernet-Like Interfaces

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

This document defines objects for managing Operations, Administration, and Maintenance (OAM) capabilities on Ethernet-like interfaces conformant to the Ethernet OAM functionality defined in the Ethernet in the First Mile (EFM) clauses of the Ethernet standards. The Ethernet OAM functionality is complementary to the Simple Network Management Protocol (SNMP) in that it is focused on a small set of link-specific functions for directly connected Ethernet interfaces. This document defines objects for controlling those link OAM functions and for providing results and status of the OAM functions to management entities.

Table of Contents

1. Introduction	2
2. The Internet-Standard Management Framework	2
3. Overview	3
3.1. Remote Fault Indication	4
3.2. Link Monitoring	4
3.3. Remote Loopback	5
3.4. Ethernet OAM Protocol Data Units	5
4. Relation to the Other MIB Modules	5
4.1. Relation to Other MIB Modules	5
4.2. Relation to Other EFM MIB Modules	6
4.3. Mapping of IEEE 802.3ah Managed Objects	6
5. MIB Structure	7
6. MIB Definition	8
7. Security Considerations	47
8. IANA Considerations	49
9. References	49
9.1. Normative References	49
9.2. Informative References	50
10. Acknowledgments	51

1. Introduction

The IEEE 802.3ah Ethernet in the First Mile (EFM) taskforce added new management capabilities to Ethernet-like interfaces. These management capabilities were introduced to provide some basic Ordered Aggregate (OA) function on Ethernet media. The defined functionality includes discovery, error signaling, loopback, and link monitoring. This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community to manage these new Ethernet interface capabilities.

Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP).

Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIV2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

3. Overview

Ethernet networks have evolved over the past 30 years from simple LANs to a variety of other applications, including wide-area networks. To address some of these emerging markets, the IEEE 802.3ah taskforce defined additional clauses in [802.3ah] for the IEEE 802.3 standard [802.3-2002] to better address Ethernet deployments in the public-access network. Although Ethernet-access deployments were the primary motivation for the taskforce activity, the results of the taskforce are not strictly limited to that application. Ethernet OAM can be implemented on Ethernet links that are not EFM.

The Ethernet in the First Mile (EFM) taskforce was focused on four somewhat independent objectives to better address Ethernet access deployments: optics, copper, Ethernet passive optical networks (Ethernet PON, or EPON), and operations, administration, and maintenance (OAM). The optics sub-taskforce developed new optical physical layers that better served the long-reach outside plant networks typically found in the access network, including developing physical layers that operate up to 20 Km and supporting the environmental conditions of access deployments. The copper sub-taskforce developed two new physical layers that run Ethernet natively over existing twisted pair wires that have been supporting voice services for decades. The EPON sub-taskforce developed a new point-to-multipoint Ethernet physical layer, utilizing Ethernet framing natively over a time-division multiple-access (TDMA) infrastructure. The OAM sub-taskforce introduced some basic management functionality into an Ethernet link to better monitor and maintain Ethernet networks in geographically disparate networks.

This document defines the management objects necessary to integrate Ethernet OAM functionality into the SNMP management framework.

Ethernet OAM is composed of a core set of functions and a set of optional functional groups. The mandatory functions include discovery operations (determining if the other end of the link is OA capable and what OAM functions it supports), state machine implementation, and some critical event flows. The optional functional groups are for (a) link events, (b) remote loopback, and (c) variable retrieval and response. Each optional functional group is controlled by a separate MIB table(s).

Ethernet OAM is complementary with SNMP management in that it provides some basic management functions at layer two, rather than using layer three and above as required by SNMP over an IP infrastructure. Ethernet OAM provides single-hop functionality in that it works only between two directly connected Ethernet stations. SNMP can be used to manage the Ethernet OAM interactions of one Ethernet station with another.

Ethernet OAM has three functional objectives, which are detailed in the next three sections. The definition of a basic Ethernet OAM protocol data unit is given in Section 3.4.

3.1. Remote Fault Indication

Remote fault indication provides a mechanism for one end of an Ethernet link to signal the other end that the receive path is non-operational. Some Ethernet physical layers offer mechanisms to signal this condition at the physical layer. Ethernet OAM added a mechanism so that some Ethernet physical layers can operate in unidirectional mode, allowing frames to be transmitted in one direction even when the other direction is non-operational. Traditionally, Ethernet PHYs do not allow frame transmission in one direction if the other direction is not operational. Using this mode, Ethernet OAM allows frame-based signaling of remote fault conditions while still not allowing higher-layer applications to be aware of the unidirectional capability. This document includes mechanisms for capturing that fault information and reflecting such information in objects and notifications within the SNMP management framework.

3.2. Link Monitoring

Ethernet OAM includes event signaling capability so that one end of an Ethernet link can indicate the occurrence of certain important events to the other end of the link. This happens via layer two protocols. This document defines methods for incorporating the occurrence of these layer two events, both at the local end and far end of the link, into the SNMP management framework.

Ethernet OAM also includes mechanisms for one Ethernet station to query another directly connected Ethernet station about the status of its Ethernet interface variables and status. This document does not include mechanisms for controlling how one Ethernet endpoint may use this functionality to query the status or statistics of a peer Ethernet entity.

3.3. Remote Loopback

Remote loopback is a link state where the peer Ethernet entity echoes every received packet (without modifications) back onto the link. Remote loopback is intrusive in that the other end of the link is not forwarding traffic from higher layers out over the link. This document defines objects controlling loopback operation and reading the status of the loopback state.

3.4. Ethernet OAM Protocol Data Units

An Ethernet OAM protocol data unit is a valid Ethernet frame with a destination Media Access Control (MAC) address equal to the reserved MAC address for Slow Protocols (See 43B of [802.3ah]), a lengthOrType field equal to the reserved type for Slow Protocols, and a Slow Protocols subtype equal to that of the subtype reserved for Ethernet OAM.

OAMPDU is used throughout this document as an abbreviation for Ethernet OAM protocol data unit. OAMPDUs are the mechanism by which two directly connected Ethernet interfaces exchange OA information.

4. Relation to the Other MIB Modules

The definitions presented here are based on Clauses 30 and 57 of [802.3ah]. Note that these clauses describe many of these variables and their effects on the MAC layer. In some cases, there is a one-to-one relationship between an object in this document and an object in the Clause 30 MIB of [802.3ah]. In other cases, the objects of this document reflect a more complex entity and are reflected by more than one object in the Clause 30 MIB of [802.3ah].

4.1. Relation to Other MIB Modules

The objects defined in this document manage OAM functionality introduced in [802.3ah]. These objects do not overlap with the interfaces MIB [RFC2863], the Ethernet-like interfaces MIB [RFC3635], or any other MIB currently used to manage various aspects of an Ethernet interface. The objects defined here are defined for Ethernet-like interfaces only and use the same ifIndex as the associated Ethernet interface. Ethernet OAM can be implemented on any Ethernet-like interface.

4.2. Relation to Other EFM MIB Modules

The Ethernet OAM functionality and MIB Module is independent of the other functionality and MIB Modules derived from [802.3ah] for copper [802.3ah-copper] and EPON [802.3ah-epon]. Ethernet OAM may be implemented (or not) on the new EFM interface types, just as it can on any other Ethernet interface.

4.3. Mapping of IEEE 802.3ah Managed Objects

This section contains the mapping between managed objects defined in [802.3ah] Clause 30, and managed objects defined in this document.

IEEE 802.3 Managed Object	Corresponding SNMP object
oOA	
.aOAMID	IF-MIB ifIndex
.aOAMAdminState	dot30amAdminState
.aOAMMode	dot30amMode
.aOAMDiscoveryState	dot30amOperStatus
.aOAMRemoteMACAddress	dot30amPeerMacAddress
.aOAMLocalConfiguration	dot30amFunctionsSupported
.aOAMRemoteConfiguration	dot30amPeerFunctionsSupported, dot30amPeerMode
.aOAMLocalPDUConfiguration	dot30amMaxOamPduSize
.aOAMRemotePDUConfiguration	dot30amPeerMaxOamPduSize
.aOAMLocalFlagsField	dot30amOperStatus, dot30amEventLogEntry
.aOAMRemoteFlagsField	dot30amOperStatus, dot30amEventLogEntry
.aOAMLocalRevision	dot30amConfigRevision
.aOAMRemoteRevision	dot30amPeerConfigRevision
.aOAMLocalState	dot30amLoopbackStatus
.aOAMRemoteState	dot30amLoopbackStatus
.aOAMRemoteVendorOUI	dot30amPeerVendorOui
.aOAMRemoteVendorSpecificInfo	dot30amPeerVendorInfo
.aOAMUnsupportedCodesTx	dot30amUnsupportedCodesTx
.aOAMUnsupportedCodesRx	dot30amUnsupportedCodesRx
.aOAMInformationTx	dot30amInformationTx
.aOAMInformationRx	dot30amInformationRx
.aOAMUniqueEventNotificationTx	dot30amUniqueEventNotificationTx
.aOAMUniqueEventNotificationRx	dot30amUniqueEventNotificationRx
.aOAMDuplicateEventNotificationTx	dot30amDuplicateEventNotificationTx
.aOAMDuplicateEventNotificationRx	dot30amDuplicateEventNotificationRx
.aOAMLoopbackControlTx	dot30amLoopbackControlTx

.aOAMLoopbackControlRx	dot30amLoopbackControlRx
.aOAMVariableRequestTx	dot30amVariableRequestTx
.aOAMVariableRequestRx	dot30amVariableRequestRx
.aOAMVariableResponseTx	dot30amVariableResponseTx
.aOAMVariableResponseRx	dot30amVariableResponseRx
.aOAMOrganizationSpecificTx	dot30amOrgSpecificTx
.aOAMOrganizationSpecificRx	dot30amOrgSpecificRx
.aOAMLocalErrSymPeriodConfig	dot30amErrSymPeriodWindow, dot30amErrSymPeriodThreshold
.aOAMLocalErrSymPeriodEvent	dot30amEventLogEntry
.aOAMLocalErrFrameConfig	dot30amErrFrameWindow, dot30amErrFrameThreshold
.aOAMLocalErrFrameEvent	dot30amEventLogEntry
.aOAMLocalErrFramePeriodConfig	dot30amErrFramePeriodWindow, dot30amErrFramePeriodThreshold
.aOAMLocalErrFramePeriodEvent	dot30amEventLogEntry
.aOAMLocalErrFrameSecsSummaryConfig	dot30amErrFrameSecsSummaryWindow, dot30amErrFrameSecssummaryThreshold
.aOAMLocalErrFrameSecsSummaryEvent	dot30amEventLogEntry
.aOAMRemoteErrSymPeriodEvent	dot30amEventLogEntry
.aOAMRemoteErrFrameEvent	dot30amEventLogEntry
.aOAMRemoteErrFramePeriodEvent	dot30amEventLogEntry
.aOAMRemoteErrFrameSecsSummaryEvent	dot30amEventLogEntry
.aFramesLostDueToOAMError	dot30amFramesLostDueToOam
.acOAMAdminControl	dot30amAdminState

There are no IEEE 802.3ah managed objects that are not reflected in this MIB Module in some manner.

5. MIB Structure

The Ethernet OAM MIB objects of this memo focus on the OA capabilities introduced in [802.3ah]. The MIB objects are partitioned into six different MIB groups.

The dot30amTable group manages the primary OAM objects of the Ethernet interface. This group controls the state and status of OA as well as the mode in which it operates.

The dot30amPeerTable maintains the current information on the status and configuration of the peer OAM entity on the Ethernet interface. Managed information includes the capabilities and function available on the peer OAM entity.

The dot3OamLoopbackTable manages the loopback function introduced in [802.3ah]. This table controls enabling and disabling loopback, as well as indicating the loopback status of Ethernet OAM on this interface.

The dot3OamStatsTable maintains statistics on the number and type of Ethernet OAM frames being transmitted and received on the Ethernet interface.

The dot3OamEventConfigTable defines the objects for managing the event notification capability available in Ethernet OAM. With Ethernet OAM, one device may send notifications to its peer devices whenever an important event happens on the local device. This table provides management of which events result in notifications via Ethernet OAM notifications and/or via SNMP notifications.

The dot3OamEventLogTable manages the current status of local and remote events detected via Ethernet OAM. This table is updated whenever local events are detected by Ethernet OAM or whenever Ethernet OAM Event Notifications are received from the peer OAM entity.

There are two notifications defined to report Ethernet OAM events (one for threshold crossing events, one for non-threshold crossing events). Both notifications are contained within the same conformance group.

6. MIB Definition

DOT3-OAM-MIB DEFINITIONS ::= BEGIN

IMPORTS

```
MODULE-IDENTITY, mib-2, OBJECT-TYPE, Counter32, Unsigned32,
    Integer32, NOTIFICATION-TYPE
    FROM SNMPv2-SMI
    -- from [RFC2578]
TEXTUAL-CONVENTION, MacAddress, TimeStamp, TruthValue

    FROM SNMPv2-TC
    -- from [RFC2579]
CounterBasedGauge64
    FROM HCNUM-TC
    -- from [RFC2856]
ifIndex
    FROM IF-MIB
    -- from [RFC2863]
MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
    FROM SNMPv2-CONF;
```


-- from [RFC2580]

dot3OamMIB MODULE-IDENTITY

LAST-UPDATED "200706140000Z" -- June 14,2007"

ORGANIZATION

"IETF Ethernet Interfaces and Hub MIB Working Group"

CONTACT-INFO

"WG Charter:

<http://www.ietf.org/html.charters/hubmib-charter.html>

Mailing lists:

General Discussion: hubmib@ietf.org

To Subscribe: hubmib-requests@ietf.org

In Body: subscribe your_email_address

Chair: Bert Wijnen

Alcatel-Lucent

Email: [bwijnen at alcatel-lucent dot com](mailto:bwijnen@alcatel-lucent.com)

Editor: Matt Squire

Hatteras Networks

E-mail: [msquire at hatterasnetworks dot com](mailto:msquire@hatterasnetworks.com)

"

DESCRIPTION

"The MIB module for managing the new Ethernet OAM features introduced by the Ethernet in the First Mile taskforce (IEEE 802.3ah). The functionality presented here is based on IEEE 802.3ah [802.3ah], released in October, 2004. [802.3ah] was prepared as an addendum to the standing version of IEEE 802.3 [802.3-2002]. Since then, [802.3ah] has been merged into the base IEEE 802.3 specification in [802.3-2005].

In particular, this MIB focuses on the new OAM functions introduced in Clause 57 of [802.3ah]. The OAM functionality of Clause 57 is controlled by new management attributes introduced in Clause 30 of [802.3ah]. The OAM functions are not specific to any particular Ethernet physical layer, and can be generically applied to any Ethernet interface of [802.3-2002].

An Ethernet OAM protocol data unit is a valid Ethernet frame with a destination MAC address equal to the reserved MAC address for Slow Protocols (See 43B of [802.3ah]), a lengthOrType field equal to the reserved type for Slow Protocols, and a Slow Protocols subtype equal to that of the subtype reserved for Ethernet OAM. OAMPDU is used throughout this document as an abbreviation for Ethernet OAM protocol data unit.

The following reference is used throughout this MIB module:

[802.3ah] refers to:

IEEE Std 802.3ah-2004: 'Draft amendment to - Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications - Media Access Control Parameters, Physical Layers and Management Parameters for subscriber access networks', October 2004.

[802.3-2002] refers to:

IEEE Std 802.3-2002: 'Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications - Media Access Control Parameters, Physical Layers and Management Parameters for subscriber access networks', March 2002.

[802.3-2005] refers to:

IEEE Std 802.3-2005: 'Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications - Media Access Control Parameters, Physical Layers and Management Parameters for subscriber access networks', December 2005.

[802-2001] refers to:

'IEEE Standard for LAN/MAN (Local Area Network/Metropolitan Area Network): Overview and Architecture', IEEE 802, June 2001.

Copyright (c) The IETF Trust (2007). This version of this MIB module is part of RFC 4878; See the RFC itself for full legal notices. "

REVISION "200706140000Z" -- June 14, 2007"
DESCRIPTION "Initial version, published as RFC 4878."
 ::= { mib-2 158 }

--

-- Sections of the Ethernet OAM MIB

```

--
dot30amNotifications OBJECT IDENTIFIER ::= { dot30amMIB 0 }
dot30amObjects        OBJECT IDENTIFIER ::= { dot30amMIB 1 }
dot30amConformance    OBJECT IDENTIFIER ::= { dot30amMIB 2 }

--
-- Textual conventions for the OAM MIB
--
EightOTwoOui ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "24-bit Organizationally Unique Identifier. Information on
        OUIs can be found in IEEE 802-2001 [802-2001], Clause 9."
    SYNTAX      OCTET STRING(SIZE(3))

-- *****
--
-- Ethernet OAM Control group
--

dot30amTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Dot30amEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table contains the primary controls and status for the
        OAM capabilities of an Ethernet-like interface. There will be
        one row in this table for each Ethernet-like interface in the
        system that supports the OAM functions defined in [802.3ah].
        "
    ::= { dot30amObjects 1 }

dot30amEntry OBJECT-TYPE
    SYNTAX      Dot30amEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry in the table that contains information on the
        Ethernet OAM function for a single Ethernet like interface.
        Entries in the table are created automatically for each
        interface supporting Ethernet OAM. The status of the row
        entry can be determined from dot30amOperStatus.

        A dot30amEntry is indexed in the dot30amTable by the ifIndex
        object of the Interfaces MIB.
        "
    INDEX      { ifIndex }
    ::= { dot30amTable 1 }

```

```

Dot30amEntry ::=
  SEQUENCE {
    dot30amAdminState          INTEGER,
    dot30amOperStatus          INTEGER,
    dot30amMode                 INTEGER,
    dot30amMaxOamPduSize        Unsigned32,
    dot30amConfigRevision       Unsigned32,
    dot30amFunctionsSupported   BITS
  }

```

dot30amAdminState OBJECT-TYPE

```

SYNTAX      INTEGER {
                enabled(1),
                disabled(2)
            }

```

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object is used to provision the default administrative OAM mode for this interface. This object represents the desired state of OAM for this interface.

The dot30amAdminState always starts in the disabled(2) state until an explicit management action or configuration information retained by the system causes a transition to the enabled(1) state. When enabled(1), Ethernet OAM will attempt to operate over this interface.

```

REFERENCE   "[802.3ah], 30.3.6.1.2"
 ::= { dot30amEntry 1 }

```

dot30amOperStatus OBJECT-TYPE

```

SYNTAX      INTEGER {
                disabled(1),
                linkFault(2),
                passiveWait(3),
                activeSendLocal(4),
                sendLocalAndRemote(5),
                sendLocalAndRemoteOk(6),
                oamPeeringLocallyRejected(7),
                oamPeeringRemotelyRejected(8),
                operational(9),
                nonOperHalfDuplex(10)
            }

```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"At initialization and failure conditions, two OAM entities on

the same full-duplex Ethernet link begin a discovery phase to determine what OAM capabilities may be used on that link. The progress of this initialization is controlled by the OA sublayer.

This value is always disabled(1) if OAM is disabled on this interface via the dot30amAdminState.

If the link has detected a fault and is transmitting OAMPDUs with a link fault indication, the value is linkFault(2). Also, if the interface is not operational (ifOperStatus is not up(1)), linkFault(2) is returned. Note that the object ifOperStatus may not be up(1) as a result of link failure or administrative action (ifAdminState being down(2) or testing(3)).

The passiveWait(3) state is returned only by OAM entities in passive mode (dot30amMode) and reflects the state in which the OAM entity is waiting to see if the peer device is OA capable. The activeSendLocal(4) value is used by active mode devices (dot30amMode) and reflects the OAM entity actively trying to discover whether the peer has OAM capability but has not yet made that determination.

The state sendLocalAndRemote(5) reflects that the local OA entity has discovered the peer but has not yet accepted or rejected the configuration of the peer. The local device can, for whatever reason, decide that the peer device is unacceptable and decline OAM peering. If the local OAM entity rejects the peer OAM entity, the state becomes oamPeeringLocallyRejected(7). If the OAM peering is allowed by the local device, the state moves to sendLocalAndRemoteOk(6). Note that both the sendLocalAndRemote(5) and oamPeeringLocallyRejected(7) states fall within the state SEND_LOCAL_REMOTE of the Discovery state diagram [802.3ah, Figure 57-5], with the difference being whether the local OAM client has actively rejected the peering or has just not indicated any decision yet. Whether a peering decision has been made is indicated via the local flags field in the OAMPDU (reflected in the aOAMLocalFlagsField of 30.3.6.1.10).

If the remote OAM entity rejects the peering, the state becomes oamPeeringRemotelyRejected(8). Note that both the sendLocalAndRemoteOk(6) and oamPeeringRemotelyRejected(8) states fall within the state SEND_LOCAL_REMOTE_OK of the Discovery state diagram [802.3ah, Figure 57-5], with the difference being whether the remote OAM client has rejected

the peering or has just not yet decided. This is indicated via the remote flags field in the OAMPDU (reflected in the aOAMRemoteFlagsField of 30.3.6.1.11).

When the local OAM entity learns that both it and the remote OAM entity have accepted the peering, the state moves to operational(9) corresponding to the SEND_ANY state of the Discovery state diagram [802.3ah, Figure 57-5].

Since Ethernet OAM functions are not designed to work completely over half-duplex interfaces, the value nonOperHalfDuplex(10) is returned whenever Ethernet OAM is enabled (dot30amAdminState is enabled(1)), but the interface is in half-duplex operation.

REFERENCE "[802.3ah], 30.3.6.1.4, 30.3.6.1.10, 30.3.6.1.11"
 ::= { dot30amEntry 2 }

dot30amMode OBJECT-TYPE

SYNTAX INTEGER {
 passive(1),
 active(2)
 }

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object configures the mode of OAM operation for this Ethernet-like interface. OAM on Ethernet interfaces may be in 'active' mode or 'passive' mode. These two modes differ in that active mode provides additional capabilities to initiate monitoring activities with the remote OAM peer entity, while passive mode generally waits for the peer to initiate OAM actions with it. As an example, an active OAM entity can put the remote OAM entity in a loopback state, where a passive OAM entity cannot.

The default value of dot30amMode is dependent on the type of system on which this Ethernet-like interface resides. The default value should be 'active(2)' unless it is known that this system should take on a subservient role to the other device connected over this interface.

Changing this value results in incrementing the configuration revision field of locally generated OAMPDUs (30.3.6.1.12) and potentially re-doing the OAM discovery process if the dot30amOperStatus was already operational(9).

REFERENCE "[802.3ah], 30.3.6.1.3"

```
::= { dot30amEntry 3 }
```

```
dot30amMaxOamPduSize OBJECT-TYPE
```

```
SYNTAX      Unsigned32 (64..1518)
```

```
UNITS       "octets"
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"The largest OAMPDU that the OAM entity supports. OA
entities exchange maximum OAMPDU sizes and negotiate to use
the smaller of the two maximum OAMPDU sizes between the peers.
This value is determined by the local implementation.
"
```

```
REFERENCE   "[802.3ah], 30.3.6.1.8"
```

```
::= { dot30amEntry 4 }
```

```
dot30amConfigRevision OBJECT-TYPE
```

```
SYNTAX      Unsigned32(0..65535)
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"The configuration revision of the OAM entity as reflected in
the latest OAMPDU sent by the OAM entity. The config revision
is used by OAM entities to indicate that configuration changes
have occurred, which might require the peer OAM entity to
re-evaluate whether OAM peering is allowed.
"
```

```
REFERENCE   "[802.3ah], 30.3.6.1.12"
```

```
::= { dot30amEntry 5 }
```

```
dot30amFunctionsSupported OBJECT-TYPE
```

```
SYNTAX      BITS {
                    unidirectionalSupport (0),
                    loopbackSupport(1),
                    eventSupport(2),
                    variableSupport(3)
                }
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"The OAM functions supported on this Ethernet-like interface.
OAM consists of separate functional sets beyond the basic
discovery process that is always required. These functional
groups can be supported independently by any implementation.
These values are communicated to the peer via the local
configuration field of Information OAMPDUs.
"
```

```
Setting 'unidirectionalSupport(0)' indicates that the OA
```

entity supports the transmission of OAMPDUs on links that are operating in unidirectional mode (traffic flowing in one direction only). Setting 'loopbackSupport(1)' indicates that the OAM entity can initiate and respond to loopback commands. Setting 'eventSupport(2)' indicates that the OAM entity can send and receive Event Notification OAMPDUs. Setting 'variableSupport(3)' indicates that the OAM entity can send and receive Variable Request and Response OAMPDUs.

```
REFERENCE    "[802.3ah], 30.3.6.1.6"
::= { dot30amEntry 6 }
```

```
-- *****
--
-- Ethernet OAM Peer group
--
```

```
dot30amPeerTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Dot30amPeerEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "This table contains information about the OAM peer for a
        particular Ethernet-like interface. OAM entities communicate
        with a single OAM peer entity on Ethernet links on which OA
        is enabled and operating properly. There is one entry in this
        table for each entry in the dot30amTable for which information
        on the peer OAM entity is available."
    ::= { dot30amObjects 2 }
```

```
dot30amPeerEntry OBJECT-TYPE
    SYNTAX      Dot30amPeerEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "An entry in the table containing information on the peer OA
        entity for a single Ethernet-like interface."

    Note that there is at most one OAM peer for each Ethernet-like
    interface. Entries are automatically created when information
    about the OAM peer entity becomes available, and automatically
    deleted when the OAM peer entity is no longer in
    communication. Peer information is not available when
    dot30amOperStatus is disabled(1), linkFault(2),
    passiveWait(3), activeSendLocal(4), or nonOperHalfDuplex(10).

    INDEX      { ifIndex }
```



```
 ::= { dot30amPeerTable 1 }
```

```
Dot30amPeerEntry ::=
```

```
SEQUENCE {
    dot30amPeerMacAddress      MacAddress,
    dot30amPeerVendorOui      Eight0Two0ui,
    dot30amPeerVendorInfo      Unsigned32,
    dot30amPeerMode            INTEGER,
    dot30amPeerMax0amPduSize   Unsigned32,
    dot30amPeerConfigRevision  Unsigned32,
    dot30amPeerFunctionsSupported BITS
}
```

```
dot30amPeerMacAddress OBJECT-TYPE
```

```
SYNTAX      MacAddress
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"The MAC address of the peer OAM entity. The MAC address is
derived from the most recently received OAMPDU.
"
```

```
REFERENCE   "[802.3ah], 30.3.6.1.5."
```

```
 ::= { dot30amPeerEntry 1 }
```

```
dot30amPeerVendorOui OBJECT-TYPE
```

```
SYNTAX      Eight0Two0ui
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"The OUI of the OAM peer as reflected in the latest
Information OAMPDU received with a Local Information TLV. The
OUI can be used to identify the vendor of the remote OA
entity. This value is initialized to three octets of zero
before any Local Information TLV is received.
"
```

```
REFERENCE   "[802.3ah], 30.3.6.1.16."
```

```
 ::= { dot30amPeerEntry 2 }
```

```
dot30amPeerVendorInfo OBJECT-TYPE
```

```
SYNTAX      Unsigned32
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"The Vendor Info of the OAM peer as reflected in the latest
Information OAMPDU received with a Local Information TLV.
The semantics of the Vendor Information field is proprietary
and specific to the vendor (identified by the
dot30amPeerVendorOui). This information could, for example,
```

be used to identify a specific product or product family. This value is initialized to zero before any Local Information TLV is received.

REFERENCE "[802.3ah], 30.3.6.1.17."
 ::= { dot30amPeerEntry 3 }

dot30amPeerMode OBJECT-TYPE

SYNTAX INTEGER {
 passive(1),
 active(2),
 unknown(3)
 }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The mode of the OAM peer as reflected in the latest Information OAMPDU received with a Local Information TLV. The mode of the peer can be determined from the Configuration field in the Local Information TLV of the last Information OAMPDU received from the peer. The value is unknown(3) whenever no Local Information TLV has been received. The values of active(2) and passive(1) are returned when a Local Information TLV has been received indicating that the peer is in active or passive mode, respectively.

REFERENCE "[802.3ah], 30.3.6.1.7."
 ::= { dot30amPeerEntry 4 }

dot30amPeerMaxOamPduSize OBJECT-TYPE

SYNTAX Unsigned32 (0 | 64..1518)

UNITS "octets"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The maximum size of OAMPDU supported by the peer as reflected in the latest Information OAMPDU received with a Local Information TLV. Ethernet OAM on this interface must not use OAMPDUs that exceed this size. The maximum OAMPDU size can be determined from the PDU Configuration field of the Local Information TLV of the last Information OAMPDU received from the peer. A value of zero is returned if no Local Information TLV has been received. Otherwise, the value of the OAM peer's maximum OAMPDU size is returned in this value.

REFERENCE "[802.3ah], 30.3.6.1.9."
 ::= { dot30amPeerEntry 5 }

dot30amPeerConfigRevision OBJECT-TYPE**SYNTAX** Unsigned32(0..65535)**MAX-ACCESS** read-only**STATUS** current**DESCRIPTION**

"The configuration revision of the OAM peer as reflected in the latest OAMPDU. This attribute is changed by the peer whenever it has a local configuration change for Ethernet OA on this interface. The configuration revision can be determined from the Revision field of the Local Information TLV of the most recently received Information OAMPDU with a Local Information TLV. A value of zero is returned if no Local Information TLV has been received.

"

REFERENCE "[802.3ah], 30.3.6.1.13."

::= { dot30amPeerEntry 6 }

dot30amPeerFunctionsSupported OBJECT-TYPE

SYNTAX BITS {
 unidirectionalSupport (0),
 loopbackSupport(1),
 eventSupport(2),
 variableSupport(3)
 }

MAX-ACCESS read-only**STATUS** current**DESCRIPTION**

"The OAM functions supported on this Ethernet-like interface. OAM consists of separate functionality sets above the basic discovery process. This value indicates the capabilities of the peer OAM entity with respect to these functions. This value is initialized so all bits are clear.

If unidirectionalSupport(0) is set, then the peer OAM entity supports sending OAM frames on Ethernet interfaces when the receive path is known to be inoperable. If loopbackSupport(1) is set, then the peer OAM entity can send and receive OAM loopback commands. If eventSupport(2) is set, then the peer OAM entity can send and receive event OAMPDUs to signal various error conditions. If variableSupport(3) is set, then the peer OAM entity can send and receive variable requests to monitor the attribute value as described in Clause 57 of [802.3ah].

The capabilities of the OAM peer can be determined from the configuration field of the Local Information TLV of the most recently received Information OAMPDU with a Local Information TLV. All zeros are returned if no Local Information TLV has

yet been received.
 "

REFERENCE "[802.3ah], REFERENCE 30.3.6.1.7."
 ::= { dot30amPeerEntry 7 }

-- *****
 --
 -- Ethernet OAM Loopback group
 --

dot30amLoopbackTable OBJECT-TYPE

SYNTAX SEQUENCE OF Dot30amLoopbackEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table contains controls for the loopback state of the local link as well as indicates the status of the loopback function. There is one entry in this table for each entry in dot30amTable that supports loopback functionality (where dot30amFunctionsSupported includes the loopbackSupport bit set)."

Loopback can be used to place the remote OAM entity in a state where every received frame (except OAMPDUs) is echoed back over the same interface on which they were received. In this state, at the remote entity, 'normal' traffic is disabled as only the looped back frames are transmitted on the interface. Loopback is thus an intrusive operation that prohibits normal data flow and should be used accordingly.
 "

::= { dot30amObjects 3 }

dot30amLoopbackEntry OBJECT-TYPE

SYNTAX Dot30amLoopbackEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in the table, containing information on the loopback status for a single Ethernet-like interface. Entries in the table are automatically created whenever the local OAM entity supports loopback capabilities. The loopback status on the interface can be determined from the dot30amLoopbackStatus object."
 "

INDEX { ifIndex }

::= { dot30amLoopbackTable 1 }

Dot30amLoopbackEntry ::=

```

SEQUENCE {
    dot30amLoopbackStatus          INTEGER,
    dot30amLoopbackIgnoreRx        INTEGER
}

```

dot30amLoopbackStatus OBJECT-TYPE

```

SYNTAX      INTEGER {
    -- all values, except where noted, can be read
    -- but cannot be written
    noLoopback (1),

    -- initiatingLoopback can be read or written
    initiatingLoopback (2),
    remoteLoopback (3),

    -- terminatingLoopback can be read or written
    terminatingLoopback (4),
    localLoopback (5),
    unknown (6)
}

```

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The loopback status of the OAM entity. This status is determined by a combination of the local parser and multiplexer states, the remote parser and multiplexer states, as well as by the actions of the local OAM client. When operating in normal mode with no loopback in progress, the status reads noLoopback(1).

The values initiatingLoopback(2) and terminatingLoopback(4) can be read or written. The other values can only be read - they can never be written. Writing initiatingLoopback causes the local OAM entity to start the loopback process with its peer. This value can only be written when the status is noLoopback(1). Writing the value initiatingLoopback(2) in any other state has no effect. When in remoteLoopback(3), writing terminatingLoopback(4) causes the local OAM entity to initiate the termination of the loopback state. Writing terminatingLoopback(4) in any other state has no effect.

If the OAM client initiates a loopback and has sent a Loopback OAMPDU and is waiting for a response, where the local parser and multiplexer states are DISCARD (see [802.3ah, 57.2.11.1]), the status is 'initiatingLoopback'. In this case, the local OAM entity has yet to receive any acknowledgment that the remote OAM entity has received its loopback command request.

If the local OAM client knows that the remote OAM entity is in loopback mode (via the remote state information as described in [802.3ah, 57.2.11.1, 30.3.6.1.15]), the status is remoteLoopback(3). If the local OAM client is in the process of terminating the remote loopback [802.3ah, 57.2.11.3, 30.3.6.1.14] with its local multiplexer and parser states in DISCARD, the status is terminatingLoopback(4). If the remote OAM client has put the local OAM entity in loopback mode as indicated by its local parser state, the status is localLoopback(5).

The unknown(6) status indicates that the parser and multiplexer combination is unexpected. This status may be returned if the OAM loopback is in a transition state but should not persist.

The values of this attribute correspond to the following values of the local and remote parser and multiplexer states.

value	LclPrsr	LclMux	RmtPrsr	RmtMux
noLoopback	FWD	FWD	FWD	FWD
initLoopback	DISCARD	DISCARD	FWD	FWD
rmtLoopback	DISCARD	FWD	LPBK	DISCARD
tmtngLoopback	DISCARD	DISCARD	LPBK	DISCARD
lclLoopback	LPBK	DISCARD	DISCARD	FWD
unknown	***	any other combination	***	***

REFERENCE "[802.3ah], REFERENCE 57.2.11, 30.3.61.14,
30.3.6.1.15"
::= { dot30amLoopbackEntry 1 }

dot30amLoopbackIgnoreRx OBJECT-TYPE

SYNTAX INTEGER { ignore(1), process(2) }
MAX-ACCESS read-write
STATUS current
DESCRIPTION

"Since OAM loopback is a disruptive operation (user traffic does not pass), this attribute provides a mechanism to provide controls over whether received OAM loopback commands are processed or ignored. When the value is ignore(1), received loopback commands are ignored. When the value is process(2), OAM loopback commands are processed. The default value is to ignore loopback commands (ignore(1))."

REFERENCE "[802.3ah], REFERENCE 57.2.11, 30.3.61.14,
30.3.6.1.15"
::= { dot30amLoopbackEntry 2 }

```
-- *****
--
-- Ethernet OAM Statistics group
--
```

dot30amStatsTable OBJECT-TYPE

SYNTAX SEQUENCE OF Dot30amStatsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table contains statistics for the OAM function on a particular Ethernet-like interface. There is an entry in the table for every entry in the dot30amTable.

The counters in this table are defined as 32-bit entries to match the counter size as defined in [802.3ah]. Given that the OA protocol is a slow protocol, the counters increment at a slow rate.

```
::= { dot30amObjects 4 }
```

dot30amStatsEntry OBJECT-TYPE

SYNTAX Dot30amStatsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in the table containing statistics information on the Ethernet OAM function for a single Ethernet-like interface. Entries are automatically created for every entry in the dot30amTable. Counters are maintained across transitions in dot30amOperStatus.

INDEX { ifIndex }

```
::= { dot30amStatsTable 1 }
```

Dot30amStatsEntry ::=

SEQUENCE {

dot30amInformationTx	Counter32,
dot30amInformationRx	Counter32,
dot30amUniqueEventNotificationTx	Counter32,
dot30amUniqueEventNotificationRx	Counter32,
dot30amDuplicateEventNotificationTx	Counter32,
dot30amDuplicateEventNotificationRx	Counter32,
dot30amLoopbackControlTx	Counter32,
dot30amLoopbackControlRx	Counter32,
dot30amVariableRequestTx	Counter32,
dot30amVariableRequestRx	Counter32,
dot30amVariableResponseTx	Counter32,

dot30amVariableResponseRx	Counter32,
dot30amOrgSpecificTx	Counter32,
dot30amOrgSpecificRx	Counter32,
dot30amUnsupportedCodesTx	Counter32,
dot30amUnsupportedCodesRx	Counter32,
dot30amFramesLostDueToOam	Counter32
}	

dot30amInformationTx OBJECT-TYPE

SYNTAX Counter32

UNITS "frames"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A count of the number of Information OAMPDUs transmitted on this interface.

Discontinuities of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of the ifCounterDiscontinuityTime. "

REFERENCE "[802.3ah], 30.3.6.1.20."

::= { dot30amStatsEntry 1 }

dot30amInformationRx OBJECT-TYPE

SYNTAX Counter32

UNITS "frames"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A count of the number of Information OAMPDUs received on this interface.

Discontinuities of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of the ifCounterDiscontinuityTime.

"

REFERENCE "[802.3ah], 30.3.6.1.21."

::= { dot30amStatsEntry 2 }

dot30amUniqueEventNotificationTx OBJECT-TYPE

SYNTAX Counter32

UNITS "frames"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A count of the number of unique Event OAMPDUs transmitted on this interface. Event Notifications may be sent in duplicate to increase the probability of successfully being received,

given the possibility that a frame may be lost in transit. Duplicate Event Notification transmissions are counted by dot30amDuplicateEventNotificationTx.

A unique Event Notification OAMPDU is indicated as an Event Notification OAMPDU with a Sequence Number field that is distinct from the previously transmitted Event Notification OAMPDU Sequence Number.

Discontinuities of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of the ifCounterDiscontinuityTime.

"

REFERENCE "[802.3ah], 30.3.6.1.22."
::= { dot30amStatsEntry 3 }

dot30amUniqueEventNotificationRx OBJECT-TYPE

SYNTAX Counter32

UNITS "frames"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A count of the number of unique Event OAMPDUs received on this interface. Event Notification OAMPDUs may be sent in duplicate to increase the probability of successfully being received, given the possibility that a frame may be lost in transit. Duplicate Event Notification receptions are counted by dot30amDuplicateEventNotificationRx.

A unique Event Notification OAMPDU is indicated as an Event Notification OAMPDU with a Sequence Number field that is distinct from the previously received Event Notification OAMPDU Sequence Number.

Discontinuities of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of the ifCounterDiscontinuityTime.

"

REFERENCE "[802.3ah], 30.3.6.1.24."
::= { dot30amStatsEntry 4 }

dot30amDuplicateEventNotificationTx OBJECT-TYPE

SYNTAX Counter32

UNITS "frames"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A count of the number of duplicate Event OAMPDUs transmitted

on this interface. Event Notification OAMPDUs may be sent in duplicate to increase the probability of successfully being received, given the possibility that a frame may be lost in transit.

A duplicate Event Notification OAMPDU is indicated as an Event Notification OAMPDU with a Sequence Number field that is identical to the previously transmitted Event Notification OAMPDU Sequence Number.

Discontinuities of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of the ifCounterDiscontinuityTime.

REFERENCE "[802.3ah], 30.3.6.1.23."
 ::= { dot30amStatsEntry 5 }

dot30amDuplicateEventNotificationRx OBJECT-TYPE

SYNTAX Counter32
 UNITS "frames"
 MAX-ACCESS read-only
 STATUS current

DESCRIPTION

"A count of the number of duplicate Event OAMPDUs received on this interface. Event Notification OAMPDUs may be sent in duplicate to increase the probability of successfully being received, given the possibility that a frame may be lost in transit.

A duplicate Event Notification OAMPDU is indicated as an Event Notification OAMPDU with a Sequence Number field that is identical to the previously received Event Notification OAMPDU Sequence Number.

Discontinuities of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of the ifCounterDiscontinuityTime.

REFERENCE "[802.3ah], 30.3.6.1.25."
 ::= { dot30amStatsEntry 6 }

dot30amLoopbackControlTx OBJECT-TYPE

SYNTAX Counter32
 UNITS "frames"
 MAX-ACCESS read-only
 STATUS current

DESCRIPTION

"A count of the number of Loopback Control OAMPDUs transmitted

on this interface.

Discontinuities of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of the ifCounterDiscontinuityTime.

REFERENCE "[802.3ah], 30.3.6.1.26."
::= { dot30amStatsEntry 7 }

dot30amLoopbackControlRx OBJECT-TYPE

SYNTAX Counter32
UNITS "frames"
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"A count of the number of Loopback Control OAMPDUs received on this interface.

Discontinuities of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of the ifCounterDiscontinuityTime.

REFERENCE "[802.3ah], 30.3.6.1.27."
::= { dot30amStatsEntry 8 }

dot30amVariableRequestTx OBJECT-TYPE

SYNTAX Counter32
UNITS "frames"
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"A count of the number of Variable Request OAMPDUs transmitted on this interface.

Discontinuities of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of the ifCounterDiscontinuityTime.

REFERENCE "[802.3ah], 30.3.6.1.28."
::= { dot30amStatsEntry 9 }

dot30amVariableRequestRx OBJECT-TYPE

SYNTAX Counter32
UNITS "frames"
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"A count of the number of Variable Request OAMPDUs received on

this interface.

Discontinuities of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of the ifCounterDiscontinuityTime.

REFERENCE "[802.3ah], 30.3.6.1.29."
::= { dot30amStatsEntry 10 }

dot30amVariableResponseTx OBJECT-TYPE

SYNTAX Counter32

UNITS "frames"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A count of the number of Variable Response OAMPDUs transmitted on this interface.

Discontinuities of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of the ifCounterDiscontinuityTime.

REFERENCE "[802.3ah], 30.3.6.1.30."
::= { dot30amStatsEntry 11 }

dot30amVariableResponseRx OBJECT-TYPE

SYNTAX Counter32

UNITS "frames"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A count of the number of Variable Response OAMPDUs received on this interface.

Discontinuities of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of the ifCounterDiscontinuityTime.

REFERENCE "[802.3ah], 30.3.6.1.31."
::= { dot30amStatsEntry 12 }

dot30amOrgSpecificTx OBJECT-TYPE

SYNTAX Counter32

UNITS "frames"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A count of the number of Organization Specific OAMPDUs

transmitted on this interface.

Discontinuities of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of the ifCounterDiscontinuityTime.

REFERENCE "[802.3ah], 30.3.6.1.32."
 ::= { dot30amStatsEntry 13 }

dot30amOrgSpecificRx OBJECT-TYPE

SYNTAX Counter32
 UNITS "frames"
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"A count of the number of Organization Specific OAMPDUs received on this interface.

Discontinuities of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of the ifCounterDiscontinuityTime.

REFERENCE "[802.3ah], 30.3.6.1.33."
 ::= { dot30amStatsEntry 14 }

dot30amUnsupportedCodesTx OBJECT-TYPE

SYNTAX Counter32
 UNITS "frames"
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"A count of the number of OAMPDUs transmitted on this interface with an unsupported op-code.

Discontinuities of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of the ifCounterDiscontinuityTime.

REFERENCE "[802.3ah], 30.3.6.1.18."
 ::= { dot30amStatsEntry 15 }

dot30amUnsupportedCodesRx OBJECT-TYPE

SYNTAX Counter32
 UNITS "frames"
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"A count of the number of OAMPDUs received on this interface

with an unsupported op-code.

Discontinuities of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of the ifCounterDiscontinuityTime.

REFERENCE "[802.3ah], 30.3.6.1.19."
 ::= { dot30amStatsEntry 16 }

dot30amFramesLostDueToOam OBJECT-TYPE

SYNTAX Counter32

UNITS "frames"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A count of the number of frames that were dropped by the OA multiplexer. Since the OAM multiplexer has multiple inputs and a single output, there may be cases where frames are dropped due to transmit resource contention. This counter is incremented whenever a frame is dropped by the OAM layer. Note that any Ethernet frame, not just OAMPDUs, may be dropped by the OAM layer. This can occur when an OAMPDU takes precedence over a 'normal' frame resulting in the 'normal' frame being dropped.

When this counter is incremented, no other counters in this MIB are incremented.

Discontinuities of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of the ifCounterDiscontinuityTime.

REFERENCE "[802.3ah], 30.3.6.1.46."
 ::= { dot30amStatsEntry 17 }

```
-- *****
--
-- Ethernet OAM Event Configuration group
--
```

dot30amEventConfigTable OBJECT-TYPE

SYNTAX SEQUENCE OF Dot30amEventConfigEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Ethernet OAM includes the ability to generate and receive Event Notification OAMPDUs to indicate various link problems. This table contains the mechanisms to enable Event

Notifications and configure the thresholds to generate the standard Ethernet OAM events. There is one entry in the table for every entry in dot30amTable that supports OAM events (where dot30amFunctionsSupported includes the eventSupport bit set). The values in the table are maintained across changes to dot30amOperStatus.

The standard threshold crossing events are:

- Errored Symbol Period Event. Generated when the number of symbol errors exceeds a threshold within a given window defined by a number of symbols (for example, 1,000 symbols out of 1,000,000 had errors).
- Errored Frame Period Event. Generated when the number of frame errors exceeds a threshold within a given window defined by a number of frames (for example, 10 frames out of 1000 had errors).
- Errored Frame Event. Generated when the number of frame errors exceeds a threshold within a given window defined by a period of time (for example, 10 frames in 1 second had errors).
- Errored Frame Seconds Summary Event. Generated when the number of errored frame seconds exceeds a threshold within a given time period (for example, 10 errored frame seconds within the last 100 seconds). An errored frame second is defined as a 1 second interval which had >0 frame errors.

There are other events (dying gasp, critical events) that are not threshold crossing events but which can be enabled/disabled via this table.

```
 ::= { dot30amObjects 5 }
```

dot30amEventConfigEntry OBJECT-TYPE

SYNTAX Dot30amEventConfigEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Entries are automatically created and deleted from this table, and exist whenever the OAM entity supports Ethernet OA events (as indicated by the eventSupport bit in dot30amFunctionsSupported). Values in the table are maintained across changes to the value of dot30amOperStatus.

Event configuration controls when the local management entity sends Event Notification OAMPDUs to its OAM peer, and when certain event flags are set or cleared in OAMPDUs.

INDEX { ifIndex }

```
 ::= { dot30amEventConfigTable 1 }
```

Dot30amEventConfigEntry ::=

```
SEQUENCE {
    dot30amErrSymPeriodWindowHi      Unsigned32,
    dot30amErrSymPeriodWindowLo      Unsigned32,
    dot30amErrSymPeriodThresholdHi   Unsigned32,
    dot30amErrSymPeriodThresholdLo   Unsigned32,
    dot30amErrSymPeriodEvNotifEnable TruthValue,
    dot30amErrFramePeriodWindow      Unsigned32,
    dot30amErrFramePeriodThreshold   Unsigned32,
    dot30amErrFramePeriodEvNotifEnable TruthValue,
    dot30amErrFrameWindow             Unsigned32,
    dot30amErrFrameThreshold          Unsigned32,
    dot30amErrFrameEvNotifEnable      TruthValue,
    dot30amErrFrameSecsSummaryWindow Integer32,
    dot30amErrFrameSecsSummaryThreshold Integer32,
    dot30amErrFrameSecsEvNotifEnable TruthValue,
    dot30amDyingGaspEnable            TruthValue,
    dot30amCriticalEventEnable        TruthValue
}
```

dot30amErrSymPeriodWindowHi OBJECT-TYPE

```
SYNTAX      Unsigned32
UNITS       "2^32 symbols"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
```

"The two objects dot30amErrSymPeriodWindowHi and dot30amErrSymPeriodLo together form an unsigned 64-bit integer representing the number of symbols over which this threshold event is defined. This is defined as

$$\text{dot30amErrSymPeriodWindow} = ((2^{32}) * \text{dot30amErrSymPeriodWindowHi}) + \text{dot30amErrSymPeriodWindowLo}$$

If dot30amErrSymPeriodThreshold symbol errors occur within a window of dot30amErrSymPeriodWindow symbols, an Event Notification OAMPDU should be generated with an Errored Symbol Period Event TLV indicating that the threshold has been crossed in this window.

The default value for dot30amErrSymPeriodWindow is the number of symbols in one second for the underlying physical layer.

```
REFERENCE   "[802.3ah], 30.3.6.1.34"
::= { dot30amEventConfigEntry 1 }
```

dot30amErrSymPeriodWindowLo OBJECT-TYPE

```
SYNTAX      Unsigned32
UNITS       "symbols"
```


MAX-ACCESS read-write
 STATUS current
 DESCRIPTION

"The two objects dot30amErrSymPeriodWindowHi and dot30amErrSymPeriodWindowLo together form an unsigned 64-bit integer representing the number of symbols over which this threshold event is defined. This is defined as

$$\text{dot30amErrSymPeriodWindow} = ((2^{32}) * \text{dot30amErrSymPeriodWindowHi}) + \text{dot30amErrSymPeriodWindowLo}$$

If dot30amErrSymPeriodThreshold symbol errors occur within a window of dot30amErrSymPeriodWindow symbols, an Event Notification OAMPDU should be generated with an Errored Symbol Period Event TLV indicating that the threshold has been crossed in this window.

The default value for dot30amErrSymPeriodWindow is the number of symbols in one second for the underlying physical layer.

REFERENCE "[802.3ah], 30.3.6.1.34"
 ::= { dot30amEventConfigEntry 2 }

dot30amErrSymPeriodThresholdHi OBJECT-TYPE

SYNTAX Unsigned32
 UNITS "2^32 symbols"
 MAX-ACCESS read-write
 STATUS current
 DESCRIPTION

"The two objects dot30amErrSymPeriodThresholdHi and dot30amErrSymPeriodThresholdLo together form an unsigned 64-bit integer representing the number of symbol errors that must occur within a given window to cause this event.

This is defined as

$$\text{dot30amErrSymPeriodThreshold} = ((2^{32}) * \text{dot30amErrSymPeriodThresholdHi}) + \text{dot30amErrSymPeriodThresholdLo}$$

If dot30amErrSymPeriodThreshold symbol errors occur within a window of dot30amErrSymPeriodWindow symbols, an Event Notification OAMPDU should be generated with an Errored Symbol Period Event TLV indicating that the threshold has been crossed in this window.

The default value for dot30amErrSymPeriodThreshold is one symbol errors. If the threshold value is zero, then an Event

Notification OAMPDU is sent periodically (at the end of every window). This can be used as an asynchronous notification to the peer OAM entity of the statistics related to this threshold crossing alarm.

REFERENCE "[802.3ah], 30.3.6.1.34"
 ::= { dot30amEventConfigEntry 3 }

dot30amErrSymPeriodThresholdLo OBJECT-TYPE

SYNTAX Unsigned32

UNITS "symbols"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The two objects dot30amErrSymPeriodThresholdHi and dot30amErrSymPeriodThresholdLo together form an unsigned 64-bit integer representing the number of symbol errors that must occur within a given window to cause this event.

This is defined as

$$\text{dot30amErrSymPeriodThreshold} = ((2^{32}) * \text{dot30amErrSymPeriodThresholdHi}) + \text{dot30amErrSymPeriodThresholdLo}$$

If dot30amErrSymPeriodThreshold symbol errors occur within a window of dot30amErrSymPeriodWindow symbols, an Event Notification OAMPDU should be generated with an Errored Symbol Period Event TLV indicating that the threshold has been crossed in this window.

The default value for dot30amErrSymPeriodThreshold is one symbol error. If the threshold value is zero, then an Event Notification OAMPDU is sent periodically (at the end of every window). This can be used as an asynchronous notification to the peer OAM entity of the statistics related to this threshold crossing alarm.

REFERENCE "[802.3ah], 30.3.6.1.34"
 ::= { dot30amEventConfigEntry 4 }

dot30amErrSymPeriodEvNotifEnable OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"If true, the OAM entity should send an Event Notification OAMPDU when an Errored Symbol Period Event occurs.

By default, this object should have the value true for Ethernet-like interfaces that support OAM. If the OAM layer does not support Event Notifications (as indicated via the dot30amFunctionsSupported attribute), this value is ignored.

```
 ::= { dot30amEventConfigEntry 5 }
```

dot30amErrFramePeriodWindow OBJECT-TYPE

SYNTAX Unsigned32

UNITS "frames"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The number of frames over which the threshold is defined. The default value of the window is the number of minimum size Ethernet frames that can be received over the physical layer in one second.

If dot30amErrFramePeriodThreshold frame errors occur within a window of dot30amErrFramePeriodWindow frames, an Event Notification OAMPDU should be generated with an Errored Frame Period Event TLV indicating that the threshold has been crossed in this window.

REFERENCE "[802.3ah], 30.3.6.1.38"

```
 ::= { dot30amEventConfigEntry 6 }
```

dot30amErrFramePeriodThreshold OBJECT-TYPE

SYNTAX Unsigned32

UNITS "frames"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The number of frame errors that must occur for this event to be triggered. The default value is one frame error. If the threshold value is zero, then an Event Notification OAMPDU is sent periodically (at the end of every window). This can be used as an asynchronous notification to the peer OAM entity of the statistics related to this threshold crossing alarm.

If dot30amErrFramePeriodThreshold frame errors occur within a window of dot30amErrFramePeriodWindow frames, an Event Notification OAMPDU should be generated with an Errored Frame Period Event TLV indicating that the threshold has been crossed in this window.

REFERENCE "[802.3ah], 30.3.6.1.38"

```
 ::= { dot30amEventConfigEntry 7 }
```

dot30amErrFramePeriodEvNotifEnable OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"If true, the OAM entity should send an Event Notification OAMPDU when an Errored Frame Period Event occurs.

By default, this object should have the value true for Ethernet-like interfaces that support OAM. If the OAM layer does not support Event Notifications (as indicated via the dot30amFunctionsSupported attribute), this value is ignored.

```
 ::= { dot30amEventConfigEntry 8 }
```

dot30amErrFrameWindow OBJECT-TYPE

SYNTAX Unsigned32

UNITS "tenths of a second"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The amount of time (in 100ms increments) over which the threshold is defined. The default value is 10 (1 second).

If dot30amErrFrameThreshold frame errors occur within a window of dot30amErrFrameWindow seconds (measured in tenths of seconds), an Event Notification OAMPDU should be generated with an Errored Frame Event TLV indicating that the threshold has been crossed in this window.

REFERENCE "[802.3ah], 30.3.6.1.36"

DEFVAL { 10 }

```
 ::= { dot30amEventConfigEntry 9 }
```

dot30amErrFrameThreshold OBJECT-TYPE

SYNTAX Unsigned32

UNITS "frames"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The number of frame errors that must occur for this event to be triggered. The default value is one frame error. If the threshold value is zero, then an Event Notification OAMPDU is sent periodically (at the end of every window). This can be used as an asynchronous notification to the peer OAM entity of the statistics related to this threshold crossing alarm.

If dot30amErrFrameThreshold frame errors occur within a window of dot30amErrFrameWindow (in tenths of seconds), an Event Notification OAMPDU should be generated with an Errored Frame Event TLV indicating the threshold has been crossed in this window.

```
REFERENCE    "[802.3ah], 30.3.6.1.36"
DEFVAL { 1 }
::= { dot30amEventConfigEntry 10 }
```

dot30amErrFrameEvNotifEnable OBJECT-TYPE

```
SYNTAX      TruthValue
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
```

"If true, the OAM entity should send an Event Notification OAMPDU when an Errored Frame Event occurs.

By default, this object should have the value true for Ethernet-like interfaces that support OAM. If the OAM layer does not support Event Notifications (as indicated via the dot30amFunctionsSupported attribute), this value is ignored.

```
DEFVAL { true }
::= { dot30amEventConfigEntry 11 }
```

dot30amErrFrameSecsSummaryWindow OBJECT-TYPE

```
SYNTAX      Integer32 (100..9000)
UNITS       "tenths of a second"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
```

"The amount of time (in 100 ms intervals) over which the threshold is defined. The default value is 100 (10 seconds).

If dot30amErrFrameSecsSummaryThreshold frame errors occur within a window of dot30amErrFrameSecsSummaryWindow (in tenths of seconds), an Event Notification OAMPDU should be generated with an Errored Frame Seconds Summary Event TLV indicating that the threshold has been crossed in this window.

```
REFERENCE    "[802.3ah], 30.3.6.1.40"
DEFVAL { 100 }
::= { dot30amEventConfigEntry 12 }
```

dot30amErrFrameSecsSummaryThreshold OBJECT-TYPE

```
SYNTAX      Integer32 (1..900)
```

UNITS "errored frame seconds"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The number of errored frame seconds that must occur for this event to be triggered. The default value is one errored frame second. If the threshold value is zero, then an Event Notification OAMPDU is sent periodically (at the end of every window). This can be used as an asynchronous notification to the peer OAM entity of the statistics related to this threshold crossing alarm.

If dot30amErrFrameSecsSummaryThreshold frame errors occur within a window of dot30amErrFrameSecsSummaryWindow (in tenths of seconds), an Event Notification OAMPDU should be generated with an Errored Frame Seconds Summary Event TLV indicating that the threshold has been crossed in this window.

REFERENCE "[802.3ah], 30.3.6.1.40"

DEFVAL { 1 }

::= { dot30amEventConfigEntry 13 }

dot30amErrFrameSecsEvNotifEnable OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"If true, the local OAM entity should send an Event Notification OAMPDU when an Errored Frame Seconds Event occurs.

By default, this object should have the value true for Ethernet-like interfaces that support OAM. If the OAM layer does not support Event Notifications (as indicated via the dot30amFunctionsSupported attribute), this value is ignored.

DEFVAL { true }

::= { dot30amEventConfigEntry 14 }

dot30amDyingGaspEnable OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"If true, the local OAM entity should attempt to indicate a dying gasp via the OAMPDU flags field to its peer OAM entity when a dying gasp event occurs. The exact definition of a dying gasp event is implementation dependent. If the system

does not support dying gasp capability, setting this object has no effect, and reading the object should always result in 'false'.

By default, this object should have the value true for Ethernet-like interfaces that support OAM. If the OAM layer does not support Event Notifications (as indicated via the dot30amFunctionsSupported attribute), this value is ignored.

```
DEFVAL { true }
::= { dot30amEventConfigEntry 15 }
```

dot30amCriticalEventEnable OBJECT-TYPE

```
SYNTAX      TruthValue
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
```

"If true, the local OAM entity should attempt to indicate a critical event via the OAMPDU flags to its peer OAM entity when a critical event occurs. The exact definition of a critical event is implementation dependent. If the system does not support critical event capability, setting this object has no effect, and reading the object should always result in 'false'.

By default, this object should have the value true for Ethernet-like interfaces that support OAM. If the OAM layer does not support Event Notifications (as indicated via the dot30amFunctionsSupported attribute), this value is ignored.

```
DEFVAL { true }
::= { dot30amEventConfigEntry 16 }
```

```
-- *****
--
-- Ethernet OAM Event Log group
--
```

dot30amEventLogTable OBJECT-TYPE

```
SYNTAX      SEQUENCE OF Dot30amEventLogEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
```

"This table records a history of the events that have occurred at the Ethernet OAM level. These events can include locally detected events, which may result in locally generated OAMPDUs, and remotely detected events, which are detected by the OAM peer entity and signaled to the local entity via

Ethernet OAM. Ethernet OAM events can be signaled by Event Notification OAMPDUs or by the flags field in any OAMPDU.

This table contains both threshold crossing events and non-threshold crossing events. The parameters for the threshold window, threshold value, and actual value (dot30amEventLogWindowXX, dot30amEventLogThresholdXX, dot30amEventLogValue) are only applicable to threshold crossing events, and are returned as all F's ($2^{32} - 1$) for non-threshold crossing events.

Entries in the table are automatically created when such events are detected. The size of the table is implementation dependent. When the table reaches its maximum size, older entries are automatically deleted to make room for newer entries.

::= { dot30amObjects 6 }

dot30amEventLogEntry OBJECT-TYPE

SYNTAX Dot30amEventLogEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in the dot30amEventLogTable. Entries are automatically created whenever Ethernet OAM events occur at the local OAM entity, and when Event Notification OAMPDUs are received at the local OAM entity (indicating that events have occurred at the peer OAM entity). The size of the table is implementation dependent, but when the table becomes full, older events are automatically deleted to make room for newer events. The table index dot30amEventLogIndex increments for each new entry, and when the maximum value is reached, the value restarts at zero.

INDEX { ifIndex, dot30amEventLogIndex }

::= { dot30amEventLogTable 1 }

Dot30amEventLogEntry ::=

SEQUENCE {

dot30amEventLogIndex	Unsigned32,
dot30amEventLogTimestamp	TimeStamp,
dot30amEventLogOui	Eight0Two0ui,
dot30amEventLogType	Unsigned32,
dot30amEventLogLocation	INTEGER,
dot30amEventLogWindowHi	Unsigned32,
dot30amEventLogWindowLo	Unsigned32,
dot30amEventLogThresholdHi	Unsigned32,


```

dot30amEventLogThresholdLo      Unsigned32,
dot30amEventLogValue            CounterBasedGauge64,
dot30amEventLogRunningTotal     CounterBasedGauge64,
dot30amEventLogEventTotal       Unsigned32
}

dot30amEventLogIndex            OBJECT-TYPE
SYNTAX      Unsigned32(1..4294967295)
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "An arbitrary integer for identifying individual events
    within the event log. "
 ::= { dot30amEventLogEntry 1 }

dot30amEventLogTimestamp        OBJECT-TYPE
SYNTAX      TimeStamp
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The value of sysUpTime at the time of the logged event. For
    locally generated events, the time of the event can be
    accurately retrieved from sysUpTime. For remotely generated
    events, the time of the event is indicated by the reception of
    the Event Notification OAMPDU indicating that the event
    occurred on the peer. A system may attempt to adjust the
    timestamp value to more accurately reflect the time of the
    event at the peer OAM entity by using other information, such
    as that found in the timestamp found of the Event Notification
    TLVs, which provides an indication of the relative time
    between events at the peer entity. "
 ::= { dot30amEventLogEntry 2 }

dot30amEventLogOui              OBJECT-TYPE
SYNTAX      EightOctetOui
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The OUI of the entity defining the object type. All IEEE
    802.3 defined events (as appearing in [802.3ah] except for the
    Organizationally Unique Event TLVs) use the IEEE 802.3 OUI of
    0x0180C2. Organizations defining their own Event Notification
    TLVs include their OUI in the Event Notification TLV that
    gets reflected here. "
 ::= { dot30amEventLogEntry 3 }

dot30amEventLogType              OBJECT-TYPE
SYNTAX      Unsigned32

```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The type of event that generated this entry in the event log. When the OUI is the IEEE 802.3 OUI of 0x0180C2, the following event types are defined:

erroredSymbolEvent(1),
 erroredFramePeriodEvent(2),
 erroredFrameEvent(3),
 erroredFrameSecondsEvent(4),
 linkFault(256),
 dyingGaspEvent(257),
 criticalLinkEvent(258)

The first four are considered threshold crossing events, as they are generated when a metric exceeds a given value within a specified window. The other three are not threshold crossing events.

When the OUI is not 71874 (0x0180C2 in hex), then some other organization has defined the event space. If event subtyping is known to the implementation, it may be reflected here. Otherwise, this value should return all F's ($2^{32} - 1$).
 "

REFERENCE "[802.3ah], 30.3.6.1.10 and 57.5.3."

::= { dot30amEventLogEntry 4 }

dot30amEventLogLocation OBJECT-TYPE

SYNTAX INTEGER { local(1), remote(2) }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Whether this event occurred locally (local(1)), or was received from the OAM peer via Ethernet OAM (remote(2)).
 "

::= { dot30amEventLogEntry 5 }

dot30amEventLogWindowHi OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"If the event represents a threshold crossing event, the two objects dot30amEventWindowHi and dot30amEventWindowLo, form an unsigned 64-bit integer yielding the window over which the value was measured for the threshold crossing event (for example, 5, when 11 occurrences happened in 5 seconds while the threshold was 10). The two objects are combined as:

$\text{dot30amEventLogWindow} = ((2^{32}) * \text{dot30amEventLogWindowHi})$
 $+ \text{dot30amEventLogWindowLo}$

Otherwise, this value is returned as all F's ($2^{32} - 1$) and adds no useful information.

"

REFERENCE "[802.3ah], 30.3.6.1.37 and 57.5.3.2."
 $::= \{ \text{dot30amEventLogEntry } 6 \}$

dot30amEventLogWindowLo OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"If the event represents a threshold crossing event, the two objects `dot30amEventWindowHi` and `dot30amEventWindowLo` form an unsigned 64-bit integer yielding the window over which the value was measured for the threshold crossing event (for example, 5, when 11 occurrences happened in 5 seconds while the threshold was 10). The two objects are combined as:

$\text{dot30amEventLogWindow} = ((2^{32}) * \text{dot30amEventLogWindowHi})$
 $+ \text{dot30amEventLogWindowLo}$

Otherwise, this value is returned as all F's ($2^{32} - 1$) and adds no useful information.

"

REFERENCE "[802.3ah], 30.3.6.1.37 and 57.5.3.2."
 $::= \{ \text{dot30amEventLogEntry } 7 \}$

dot30amEventLogThresholdHi OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"If the event represents a threshold crossing event, the two objects `dot30amEventThresholdHi` and `dot30amEventThresholdLo` form an unsigned 64-bit integer yielding the value that was crossed for the threshold crossing event (for example, 10, when 11 occurrences happened in 5 seconds while the threshold was 10). The two objects are combined as:

$\text{dot30amEventLogThreshold} = ((2^{32}) * \text{dot30amEventLogThresholdHi})$
 $+ \text{dot30amEventLogThresholdLo}$

Otherwise, this value is returned as all F's ($2^{32} - 1$) and adds no useful information.

"

REFERENCE "[802.3ah], 30.3.6.1.37 and 57.5.3.2."
 ::= { dot30amEventLogEntry 8 }

dot30amEventLogThresholdLo OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"If the event represents a threshold crossing event, the two objects dot30amEventThresholdHi and dot30amEventThresholdLo form an unsigned 64-bit integer yielding the value that was crossed for the threshold crossing event (for example, 10, when 11 occurrences happened in 5 seconds while the threshold was 10). The two objects are combined as:

$$\text{dot30amEventLogThreshold} = ((2^{32}) * \text{dot30amEventLogThresholdHi}) + \text{dot30amEventLogThresholdLo}$$

Otherwise, this value is returned as all F's ($2^{32} - 1$) and adds no useful information.
 "

REFERENCE "[802.3ah], 30.3.6.1.37 and 57.5.3.2."
 ::= { dot30amEventLogEntry 9 }

dot30amEventLogValue OBJECT-TYPE

SYNTAX CounterBasedGauge64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"If the event represents a threshold crossing event, this value indicates the value of the parameter within the given window that generated this event (for example, 11, when 11 occurrences happened in 5 seconds while the threshold was 10).

Otherwise, this value is returned as all F's ($2^{64} - 1$) and adds no useful information.
 "

REFERENCE "[802.3ah], 30.3.6.1.37 and 57.5.3.2."
 ::= { dot30amEventLogEntry 10 }

dot30amEventLogRunningTotal OBJECT-TYPE

SYNTAX CounterBasedGauge64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Each Event Notification TLV contains a running total of the number of times an event has occurred, as well as the number of times an Event Notification for the event has been

transmitted. For non-threshold crossing events, the number of events (dot30amLogRunningTotal) and the number of resultant Event Notifications (dot30amLogEventTotal) should be identical.

For threshold crossing events, since multiple occurrences may be required to cross the threshold, these values are likely different. This value represents the total number of times this event has happened since the last reset (for example, 3253, when 3253 symbol errors have occurred since the last reset, which has resulted in 51 symbol error threshold crossing events since the last reset).

REFERENCE "[802.3ah], 30.3.6.1.37 and 57.5.3.2."
 ::= { dot30amEventLogEntry 11 }

dot30amEventLogEventTotal OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Each Event Notification TLV contains a running total of the number of times an event has occurred, as well as the number of times an Event Notification for the event has been transmitted. For non-threshold crossing events, the number of events (dot30amLogRunningTotal) and the number of resultant Event Notifications (dot30amLogEventTotal) should be identical.

For threshold crossing events, since multiple occurrences may be required to cross the threshold, these values are likely different. This value represents the total number of times one or more of these occurrences have resulted in an Event Notification (for example, 51 when 3253 symbol errors have occurred since the last reset, which has resulted in 51 symbol error threshold crossing events since the last reset).

REFERENCE "[802.3ah], 30.3.6.1.37 and 57.5.3.2."
 ::= { dot30amEventLogEntry 12 }

```
-- *****
--
-- Ethernet OAM Notifications
--
```

dot30amThresholdEvent NOTIFICATION-TYPE

OBJECTS { dot30amEventLogTimestamp,
 dot30amEventLogOui,

```

dot30amEventLogType,
dot30amEventLogLocation,
dot30amEventLogWindowHi,
dot30amEventLogWindowLo,
dot30amEventLogThresholdHi,
dot30amEventLogThresholdLo,
dot30amEventLogValue,
dot30amEventLogRunningTotal,
dot30amEventLogEventTotal

```

```

}

```

STATUS current

DESCRIPTION

"A dot30amThresholdEvent notification is sent when a local or remote threshold crossing event is detected. A local threshold crossing event is detected by the local entity, while a remote threshold crossing event is detected by the reception of an Ethernet OAM Event Notification OAMPDU that indicates a threshold event.

This notification should not be sent more than once per second.

The OAM entity can be derived from extracting the ifIndex from the variable bindings. The objects in the notification correspond to the values in a row instance in the dot30amEventLogTable.

The management entity should periodically check dot30amEventLogTable to detect any missed events."

```

::= { dot30amNotifications 1 }

```

dot30amNonThresholdEvent NOTIFICATION-TYPE

```

OBJECTS { dot30amEventLogTimestamp,
dot30amEventLogOui,
dot30amEventLogType,
dot30amEventLogLocation,
dot30amEventLogEventTotal
}

```

STATUS current

DESCRIPTION

"A dot30amNonThresholdEvent notification is sent when a local or remote non-threshold crossing event is detected. A local event is detected by the local entity, while a remote event is detected by the reception of an Ethernet OAM Event Notification OAMPDU that indicates a non-threshold crossing event.

This notification should not be sent more than once per

second.

The OAM entity can be derived from extracting the ifIndex from the variable bindings. The objects in the notification correspond to the values in a row instance of the dot30amEventLogTable.

The management entity should periodically check dot30amEventLogTable to detect any missed events."
 ::= { dot30amNotifications 2 }

-- *****

--

-- Ethernet OAM Compliance group

--

dot30amGroups OBJECT IDENTIFIER ::= { dot30amConformance 1 }
 dot30amCompliances OBJECT IDENTIFIER ::= { dot30amConformance 2 }

-- Compliance statements

dot30amCompliance MODULE-COMPLIANCE

STATUS current

DESCRIPTION "The compliance statement for managed entities
 supporting OAM on Ethernet-like interfaces."
 "

MODULE -- this module

MANDATORY-GROUPS { dot30amControlGroup,
 dot30amPeerGroup,
 dot30amStatsBaseGroup
 }

GROUP dot30amLoopbackGroup

DESCRIPTION

"This group is mandatory for all IEEE 802.3 OA
 implementations that support loopback functionality. "

GROUP dot30amErrSymbolPeriodEventGroup

DESCRIPTION

"This group is mandatory for all IEEE 802.3 OA
 implementations that support event functionality. "

GROUP dot30amErrFramePeriodEventGroup

DESCRIPTION

"This group is mandatory for all IEEE 802.3 OA
 implementations that support event functionality. "

GROUP dot30amErrFrameEventGroup

DESCRIPTION

"This group is mandatory for all IEEE 802.3 OA implementations that support event functionality. "

GROUP dot30amErrFrameSecsSummaryEventGroup

DESCRIPTION

"This group is mandatory for all IEEE 802.3 OA implementations that support event functionality. "

GROUP dot30amFlagEventGroup

DESCRIPTION

"This group is optional for all IEEE 802.3 OA implementations. The ability to send critical events or dying gasp events is not required in any system."

GROUP dot30amEventLogGroup

DESCRIPTION

"This group is optional for all IEEE 802.3 OA implementations. Entries in this table are dependent on what event functionality is supported in the local OA implementation. At least one type of event must be supported for entries to appear in this table. "

GROUP dot30amNotificationGroup

DESCRIPTION

"This group is optional for all IEEE 802.3 OA implementations. Since the information in the notifications is dependent on the dot30amEventLogTable, that table must be implemented for notifications. "

::= { dot30amCompliances 1 }

dot30amControlGroup OBJECT-GROUP

OBJECTS { dot30amAdminState,
dot30amOperStatus,
dot30amMode,
dot30amMaxOamPduSize,
dot30amConfigRevision,
dot30amFunctionsSupported
}

STATUS current

DESCRIPTION

"A collection of objects providing the abilities, configuration, and status of an Ethernet OAM entity. "

::= { dot30amGroups 1 }

dot30amPeerGroup OBJECT-GROUP

OBJECTS { dot30amPeerMacAddress,


```

        dot30amPeerVendorOui,
        dot30amPeerVendorInfo,
        dot30amPeerMode,
        dot30amPeerFunctionsSupported,
        dot30amPeerMaxOamPduSize,
        dot30amPeerConfigRevision
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing the abilities,
        configuration, and status of a peer Ethernet OAM entity.  "
    ::= { dot30amGroups 2 }

dot30amStatsBaseGroup OBJECT-GROUP
    OBJECTS      {
        dot30amInformationTx,
        dot30amInformationRx,
        dot30amUniqueEventNotificationTx,
        dot30amUniqueEventNotificationRx,
        dot30amDuplicateEventNotificationTx,
        dot30amDuplicateEventNotificationRx,
        dot30amLoopbackControlTx,
        dot30amLoopbackControlRx,
        dot30amVariableRequestTx,
        dot30amVariableRequestRx,
        dot30amVariableResponseTx,
        dot30amVariableResponseRx,
        dot30amOrgSpecificTx,
        dot30amOrgSpecificRx,
        dot30amUnsupportedCodesTx,
        dot30amUnsupportedCodesRx,
        dot30amFramesLostDueToOam
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing the statistics for the
        number of various transmit and receive events for OAM on an
        Ethernet-like interface.  Note that all of these counters must
        be supported even if the related function (as described in
        dot30amFunctionsSupported) is not supported.  "
    ::= { dot30amGroups 3 }

dot30amLoopbackGroup OBJECT-GROUP
    OBJECTS      {
        dot30amLoopbackStatus,
        dot30amLoopbackIgnoreRx
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects for controlling the OAM remote

```

```

    loopback function.  "
    ::= { dot30amGroups 4 }

```

```

dot30amErrSymbolPeriodEventGroup OBJECT-GROUP
OBJECTS      {      dot30amErrSymPeriodWindowHi,
                    dot30amErrSymPeriodWindowLo,
                    dot30amErrSymPeriodThresholdHi,
                    dot30amErrSymPeriodThresholdLo,
                    dot30amErrSymPeriodEvNotifEnable
                }

```

```

STATUS      current

```

```

DESCRIPTION

```

"A collection of objects for configuring the thresholds for an Errored Symbol Period Event.

Each [802.3ah] defined Event Notification TLV has its own conformance group because each event can be implemented independently of any other. "

```

    ::= { dot30amGroups 5 }

```

```

dot30amErrFramePeriodEventGroup OBJECT-GROUP
OBJECTS      {      dot30amErrFramePeriodWindow,
                    dot30amErrFramePeriodThreshold,
                    dot30amErrFramePeriodEvNotifEnable
                }

```

```

STATUS      current

```

```

DESCRIPTION

```

"A collection of objects for configuring the thresholds for an Errored Frame Period Event.

Each [802.3ah] defined Event Notification TLV has its own conformance group because each event can be implemented independently of any other. "

```

    ::= { dot30amGroups 6 }

```

```

dot30amErrFrameEventGroup OBJECT-GROUP
OBJECTS      {      dot30amErrFrameWindow,
                    dot30amErrFrameThreshold,
                    dot30amErrFrameEvNotifEnable
                }

```

```

STATUS      current

```

```

DESCRIPTION

```

"A collection of objects for configuring the thresholds for an Errored Frame Event.

Each [802.3ah] defined Event Notification TLV has its own conformance group because each event can be implemented independently of any other. "

```
::= { dot30amGroups 7 }
```

```
dot30amErrFrameSecsSummaryEventGroup OBJECT-GROUP
OBJECTS { dot30amErrFrameSecsSummaryWindow,
           dot30amErrFrameSecsSummaryThreshold,
           dot30amErrFrameSecsEvNotifEnable
        }
```

```
STATUS current
```

```
DESCRIPTION
```

```
"A collection of objects for configuring the thresholds for an
Errored Frame Seconds Summary Event."
```

```
Each [802.3ah] defined Event Notification TLV has its own
conformance group because each event can be implemented
independently of any other. "
```

```
::= { dot30amGroups 8 }
```

```
dot30amFlagEventGroup OBJECT-GROUP
OBJECTS { dot30amDyingGaspEnable,
           dot30amCriticalEventEnable
        }
```

```
STATUS current
```

```
DESCRIPTION
```

```
"A collection of objects for configuring the sending OAMPDUs
with the critical event flag or dying gasp flag enabled. "
```

```
::= { dot30amGroups 9 }
```

```
dot30amEventLogGroup OBJECT-GROUP
OBJECTS { dot30amEventLogTimestamp,
           dot30amEventLogOui,
           dot30amEventLogType,
           dot30amEventLogLocation,
           dot30amEventLogWindowHi,
           dot30amEventLogWindowLo,
           dot30amEventLogThresholdHi,
           dot30amEventLogThresholdLo,
           dot30amEventLogValue,
           dot30amEventLogRunningTotal,
           dot30amEventLogEventTotal
        }
```

```
STATUS current
```

```
DESCRIPTION
```

```
"A collection of objects for configuring the thresholds for an
Errored Frame Seconds Summary Event and maintaining the event
information. "
```

```
::= { dot30amGroups 10 }
```

```
dot30amNotificationGroup NOTIFICATION-GROUP
```

```
NOTIFICATIONS {
    dot30amThresholdEvent,
    dot30amNonThresholdEvent
}
STATUS          current
DESCRIPTION
    "A collection of notifications used by Ethernet OAM to signal
    to a management entity that local or remote events have
    occurred on a specified Ethernet link. "
 ::= { dot30amGroups 11 }

END
```

7. Security Considerations

The readable objects in this module can provide information about network traffic, and therefore may be considered sensitive. In particular, OAM provides mechanisms for reading the IEEE 802.3 Clause 30 MIB attributes from a link partner via a specialized layer two protocol. Unlike SNMP, IEEE P802.3ah OAM does not include encryption or authentication mechanisms. It should be used in environments where either this interface information is not considered sensitive, or where the facility terminations are protected. By default, OAM is disabled on Ethernet-like interfaces and is therefore not a risk.

IEEE 802.3ah OAM is designed to support deployment in access and enterprise networks. In access networks, one end of a link is the CO-side, and the other is the CPE-side, and the facilities are often protected in wiring cages or closets. In such deployments, it is often the case that the CO-side is protected from access from the CPE-side. Within IEEE P802.3ah OAM, this protection from remote access is accomplished by configuring the CPE-side in passive mode using the dot30amMode attribute. This prevents the CPE from accessing functions and information at the CO-side of the connection. In enterprise networks, read-only interface information is often considered non-sensitive.

The frequency of OAM PDUs on an Ethernet interface does not adversely affect data traffic, as OAM is a slow protocol with very limited bandwidth potential, and it is not required for normal link operation. And although there are a number of objects in this module with read-write or read-create MAX-ACCESS, they have limited effects on user data.

The loopback capability of OAM can have potentially disruptive effects in that when enabling remote loopback, the remote station automatically transmits all received traffic back to the local station except for OAM traffic. This completely disrupts all higher

layer protocols such as bridging, IP, and SNMP. Therefore an attribute (dot30amLoopbackIgnoreRx) was introduced to control whether the local station processes or ignores received loopback commands.

The administrative state and mode are also read-write objects. Disabling OAM can interrupt management activities between peer devices, potentially causing serious problems. Setting the dot30amMode to an undesired value can allow access to Ethernet monitoring, events, and functions that may not be acceptable in a particular deployment scenario. In addition to loopback functionality, Ethernet interface statistics and events can be accessed via the OAM protocol, which may not be desired in some circumstances.

OAM event configuration also contains read-write objects. These objects control whether events are sent, and at what thresholds. Note that the frequency of event communication is limited by the frequency limits of Slow Protocols on Ethernet interfaces. Also, the information available via OAM events is also available via OA Variable Requests. Access to this information via either OAM events or Variable Requests is controlled by the dot30amAdminState and dot30amMode objects. As mentioned previously, inadequate protection of these variables can result in access to link information and functions.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

8. IANA Considerations

The Ethernet OAM MIB requires the allocation of a single object identifier for its MODULE-IDENTITY under the MIB-2 tree.

The MIB module in this document uses the following IANA-assigned OBJECT IDENTIFIER values recorded in the SMI Numbers registry:

Descriptor	OBJECT IDENTIFIER
-----	-----
dot30amMIB	{ mib-2 158 }

9. References

9.1. Normative References

- [802.3ah] Institute of Electrical and Electronic Engineers, IEEE Std 802.3ah-2004, "Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications - Amendment: Media Access Control Parameters, Physical Layers and Management Parameters for Subscriber Access Networks", October 2004.
- [802.3-2002] Institute of Electrical and Electronic Engineers, IEEE Std 802.3-2003, "IEEE Standard for Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications - Draft amendment to - Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications - Media Access Control Parameters, Physical Layers and Management Parameters", March 2002.

- [802.3-2005] Institute of Electrical and Electronic Engineers, IEEE Std 802.3-2005, "IEEE Standard for Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications - Draft amendment to - Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications - Media Access Control Parameters, Physical Layers and Management Parameters", December 2005.
- [802-2001] Institute of Electrical and Electronic Engineers, IEEE Std 802-2001, "Standard for Local and Metropolitan Area Networks: Architecture and Overview", March 2002.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2578] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.
- [RFC2579] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.
- [RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.
- [RFC2856] Bierman, A., McCloghrie, K., and R. Presuhn, "Textual Conventions for Additional High Capacity Data Types", RFC 2856, June 2000.
- [RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.

9.2. Informative References

- [802.3ah-copper] Beili, Ed, "Ethernet in the First Mile Copper (EFMCu) Interfaces MIB", Work in Progress, February 2007.

- [802.3ah-epon] Khermish, L., "Managed Objects of Ethernet Passive Optical Networks (EPON)", RFC 4837, June 2007.
- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.
- [RFC3635] Flick, J., "Definitions of Managed Objects for the Ethernet-like Interface Types", RFC 3635, September 2003.

10. Acknowledgments

The author is grateful to all of the participants in the IEEE 802.3ah EFM (Ethernet in the First Mile) taskforce. In particular, the strong leadership and dedication of the following individuals is noted:

Kevin Daines (Editor, IEEE 802.3ah OAM clauses)
Ben Brown (Editor, IEEE 802.3ah Logic clauses)
David Law (Editor, IEEE 802.3ah Management clauses)
Scott Simon (Editor, IEEE 802.3ah Clause 45)
Howard Frazier (Chair, IEEE 802.3ah)
Hugh Barass (Vice-Chair, IEEE 802.3ah)
Wael Diab (Editor, IEEE 802.3ah)

Additionally, certain devoted attendees and contributors to the IEEE 802.3ah OAM sub-taskforce deserve recognition. Although there were many contributors, the following individuals contributed heavily over a long period of time.

Brian Arnold
Brad Booth
Al Braga
Floyd Gerhardt
Bob Grow
Eric Lynskey
David Martin
John Messenger
Dan Romascanu (Ex-Chair, IETF HUBMIB WG)
Jonathan Thatcher
Geoff Thompson

Author's Address

**Matt Squire
Hatteras Networks
529 Davis Drive
Durham, NC 27713
EMail: msquire@hatterasnetworks.com**

Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.