

## Ethernet in the First Mile (EFM) OAM MIB

### Status of this Memo

By submitting this Internet-Draft, I certify that any applicable patent or other IPR claims of which I am aware have been disclosed, or will be disclosed, and any of which I become aware will be disclosed, in accordance with [RFC 3668](#) [[RFC3668](#)].

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC 2026](#) [[RFC2026](#)].

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

### Abstract

This document defines objects for managing Operations, Administration, and Maintenance (OAM) capabilities on Ethernet like Ethernet like interfaces conformant to the Ethernet OAM functionality defined in [[802.3ah](#)]. The Ethernet OAM functionality is complementary to SNMP management in that it is focused on a small set of link-specific functions for Ethernet interfaces. This document defines objects for controlling those link OAM functions, and on providing mechanisms to take status and input from Ethernet OAM and feed it into a larger TCP/IP network management system.

### 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](#)].

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction.....</a>	<a href="#">2</a>
<a href="#">2.</a>	<a href="#">The Internet-Standard Management Framework.....</a>	<a href="#">2</a>
<a href="#">3.</a>	<a href="#">Overview.....</a>	<a href="#">3</a>
	<a href="#">3.1 Remote fault indication.....</a>	<a href="#">4</a>
	<a href="#">3.2 Link monitoring.....</a>	<a href="#">4</a>
	<a href="#">3.3 Remote loopback.....</a>	<a href="#">4</a>
<a href="#">4.</a>	<a href="#">Relation to the Other MIBs.....</a>	<a href="#">5</a>
	<a href="#">4.1 Relation to other SNMP MIBs.....</a>	<a href="#">5</a>
	<a href="#">4.2 Relation to other EFM MIBs.....</a>	<a href="#">5</a>
	<a href="#">4.3 IANA Considerations.....</a>	<a href="#">6</a>
	<a href="#">4.4 Mapping of IEEE 802.3ah Managed Objects.....</a>	<a href="#">6</a>
<a href="#">5.</a>	<a href="#">MIB Structure.....</a>	<a href="#">7</a>
<a href="#">6.</a>	<a href="#">MIB Definition.....</a>	<a href="#">8</a>
<a href="#">7.</a>	<a href="#">Security Considerations.....</a>	<a href="#">49</a>
<a href="#">8.</a>	<a href="#">References.....</a>	<a href="#">50</a>
	<a href="#">8.1 Normative References.....</a>	<a href="#">50</a>
	<a href="#">8.2 Informative References.....</a>	<a href="#">51</a>
<a href="#">9.</a>	<a href="#">Acknowledgments.....</a>	<a href="#">52</a>
<a href="#">10.</a>	<a href="#">Author's Address.....</a>	<a href="#">52</a>
<a href="#">11.</a>	<a href="#">Intellectual Property Statement.....</a>	<a href="#">53</a>
<a href="#">12.</a>	<a href="#">Copyright Statement.....</a>	<a href="#">53</a>

## [1. Introduction](#)

The IEEE 802.3ah Ethernet in the First Mile (EFM) task force added new management capabilities to Ethernet like interfaces. These management capabilities were introduced to provide some basic OAM 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 EFM interface capabilities.

## [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 task force defined additional clauses for the IEEE 802.3 standard [[802.3-2002](#)] to better address Ethernet deployments in the public access network.

The Ethernet in the First Mile (EFM) task force 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 20Km and supporting the environmental conditions of outside 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 that are not required to be implemented along with the core set. The mandatory functions include discovery operations (determining if the other end of the link is OAM capable, and what OAM functions it implements/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, not competitive, 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 following sections.

### **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 into 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 a layer two protocol. 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.

#### **4. Relation to the Other MIBs**

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 affects 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 objectx in the Clause 30 MIB of [\[802.3ah\]](#).

##### **4.1 Relation to other SNMP MIBs**

This objects defined in this document do not overlap with MIB-2 [\[RFC1213\]](#), the interfaces MIB [\[RFC2863\]](#), or the Ethernet-likeEthernet like interfaces MIB [\[RFC3635\]](#). The objects defined here are defined for Ethernet-likeEthernet like interfaces only and use the same ifIndex as the associated Ethernet interface. Ethernet OAM can be implemented on any Ethernet like interface managed via these MIBs.

##### **4.2 Relation to other EFM MIBs**

ThisThe Ethernet OAM functionality and MIB document is independent of the other functionality and MIBs derived from [\[802.3ah\]](#) for copper [\[802.3ah-copper\]](#) and EPON [\[802.3ah-epon\]](#).

Ethernet OAM may be implemented on point-to-multipoint EFM EPON interfaces. However, because higher layer protocols that run over Ethernet interfaces are not designed for the partial connectivity provided by a point-to-multipoint interface, EPON provides a point to-point emulation layer (see [\[802.3ah\]](#) and [\[802.3ah-epon\]](#)) whereby the single EPON interface of 1-to-N connectivity is represented via N point-to-point interfaces. Ethernet OAM, like any other protocol at the Ethernet layer or above (for example, bridging), utilizes the point-to-point emulation layer of EPON in that the EPON interface is viewed as N point-to-point Ethernet interfaces. Thus OAM, and other protocols, do not need to be altered for the EPON environment.

Ethernet OAM may be implemented on the 2BASE-TL and 10PASS-TS Ethernet-over-copper interfaces defined in EFM [\[802.3ah\]](#). 2BASE-TL and 10PASS-TS can be aggregated interfaces, meaning that they can use the ifStackTable of the Interfaces Group MIB [\[RFC2863\]](#) to manage a set of N ( $1 \leq N \leq 32$ ) physical layers into a single Ethernet interface.





The other Ethernet interfaces introduced in EFM [802.3ah] are simply new optical physical layers that are managed by minimal extensions to the MAU MIB [RFC3636] defining new types of Ethernet interfaces.

#### **4.24.3 IANA Considerations**

The EFM OAM MIB requires the allocation of a single object identifier for its MODULE-IDENTITY under the MIB-2 tree. IANA has not yet allocated this object identifier.

#### **4.34.4 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
.aOAMID	IF-MIB ifIndex
.aOAMAdminState	dot30amAdminState
.aOAMMode	dot30amMode
.aOAMDiscoveryState	dot30amOperStatus
.aOAMRemoteMACAddress	dot30amPeerMacAddress
.aOAMLocalConfiguration	dot30amFunctionsSupported
.aOAMRemoteConfiguration	dot30amPeerFunctionsSupported, dot30amPeerMode
.aOAMLocalPDUConfiguration	dot30amMaxOamPduSize
.aOAMRemotePDUConfiguration	dot30amPeerMaxOamPduSize
.aOAMLocalFlagsField	dot30amOperStatus, dot30amLclErrEventFlagsData
.aOAMRemoteFlagsField	dot30amOperStatus, dot30amRmtErrEventFlagsData
.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



```

.a0AMLoopbackControlRx      dot30amLoopbackControlRx
.a0AMVariableRequestTx      dot30amVariableRequestTx
.a0AMVariableRequestRx      dot30amVariableRequestRx
.a0AMVariableResponseTx     dot30amVariableResponseTx
.a0AMVariableResponseRx     dot30amVariableResponseRx
.a0AMOrganizationSpecificTx dot30amOrgSpecificTx
.a0AMOrganizationSpecificRx dot30amOrgSpecificTx

.a0AMLocalErrSymPeriodConfig dot30amErrSymPeriodWindow,
                             dot30amErrSymPeriodThreshold

.a0AMLocalErrSymPeriodEvent
  dot30amLclErrSymPeriodData dot30amEventLogEntry
.a0AMLocalErrFrameConfig     dot30amErrFrameWindow,
                             dot30amErrFrameThreshold

.a0AMLocalErrFrameEvent
  dot30amEventLogEntry dot30amLclErrFrameData
.a0AMLocalErrFramePeriodConfig dot30amErrFramePeriodWindow,
                             dot30amErrFramePeriodThreshold

.a0AMLocalErrFramePeriodEven
  dot30amEventLogEntry dot30amLclErrFramePeriodData
.a0AMLocalErrFrameSecsSummaryConfig
                             dot30amErrFrameSecsSummaryWindow,
                             dot30amErrFrameSecsSummaryThreshold

.a0AMLocalErrFrameSecsSummaryEvent

  dot30amEventLogEntry dot30amLclErrFrameSecsSumData
.a0AMRemoteErrSymPeriodEvent
  dot30amEventLogEntry dot30amRmtErrSymPeriodData
.a0AMRemoteErrFrameEvent
  dot30amEventLogEntry dot30amRmtErrFrameData
.a0AMRemoteErrFramePeriodEven
  dot30amEventLogEntry dot30amRmtErrFramePeriodData
.a0AMRemoteErrFrameSecsSummaryEvent

  dot30amEventLogEntry dot30amRmtErrFrameSecsSumData
.aFramesLostDueToOamError
.acOamAdminControl          dot30amFramesLostDueToOam

```

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

## 5. MIB Structure

The common EFM MIB objects of this memo focus on the OAM capabilities

introduced in IEEE 802.3ah. The MIB objects are partitioned into four (4) six different MIB groups.

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

The dot30amStats table maintains statistics on the number and type of Ethernet OAM frames being transmitted and received on the Ethernet interface.

The dot30amPeerT table 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 dot30amLoopbackTable manages the loopback function introduced in EFM [802.3ah]. This table controls enabling and disabling loopback, as well as indicating the loopback status of Ethernet OAM on this interface.

The dot30amStatsTable maintains statistics on the number and type of Ethernet OAM frames being transmitted and received on the Ethernet interface.

The dot30amEventConfigTable dot30amEvent table defines the management objects for managing the event notification capability available in IEEE P802.3ah Ethernet OAM. With IEEE P802.3ah EFM 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 EFM OAM notifications and via SNMP notifications.

The dot30amEventLogTable 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 EFM OAM Event Notifications are received from the peer OAM entity.

## 6. MIB Definition

```
EFM-COMMON-MIB DEFINITIONS ::= BEGIN
  IMPORTS
    MODULE-IDENTITY, mib-2, OBJECT-TYPE, Counter32, Unsigned32,
      Integer32, NOTIFICATION-TYPE
    FROM SNMPv2-SMI
    TEXTUAL-CONVENTION, MacAddress, DateAndTime
    FROM SNMPv2-TC
    CounterBasedGauge64
    FROM HCNM-TC
```



ifIndex

FROM IF-MIB  
MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP  
FROM SNMPv2-CONF;

efmOamMIB MODULE-IDENTITY

LAST-UPDATED "200410240000Z" -- October 24, 2004"

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: Dan Romascanu, Avaya

Tel: +972-3-645-8414

Email: dromasca@avaya.com

Editor: Matt Squire

Hatteras Networks

Tel: +1-919-991-5460

Fax: +1-919-991-0743

E-mail: msquire@hatterasnetworks.com

"

DESCRIPTION

"The MIB module for managing the new Ethernet OAM features introduced by the Ethernet in the First Mile task force (IEEE 802.3ah). The functionality presented here is based on IEEE 802.3ah [[802.3ah](#)], released in October, 2004.

In particular, this MIB focused on the changes to Clause 30

of

the draft that are not specific to any physical layer. These changes are primarily reflected in the new OAM features developed under this project, that can be applied to any Ethernet like interface. The OAM features are described in Clause 57 of [[802.3ah](#)].

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.

[802-2001] refers to:

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

Copyright (c) The Internet Society (2004). This version of  
this MIB module is part of RFC XXXX; See the RFC itself for  
full legal notices. "

-- RFC Editor: Update XXXX to appropriate RFC number  
-- RFC Editor: Remove these notes

REVISION "200410240000Z" -- October 24, 2004"  
DESCRIPTION "Initial version, published as RFC XXXX."  
-- RFC Editor: Update XXXX to appropriate RFC number  
-- RFC Editor: Remove these notes

::= { mib-2 XXX }  
-- RFC Editor: Replace value with IANA assigned number  
-- RFC Editor: Remove these notes

--

-- Sections of the EFM OAM MIB

--

dot30amMIB OBJECT IDENTIFIER ::= { efmOamMIB 1 }  
dot30amConformance OBJECT IDENTIFIER ::= { efmOamMIB 2 }

--

-- Textual conventions for OAM MIB

--

Dot30ui ::= 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
    "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 Ethernet OAM functions defined in [802.3ah]."
  ::= { dot30amMIB 1 }
```

```
dot30amEntry OBJECT-TYPE
  SYNTAX      Dot30amEntry
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "An entry in the table, containing information on the
    Ethernet
    OAM function for a single Ethernet like interface."
  INDEX       { ifIndex }
  ::= { dot30amTable 1 }
```

```
Dot30amEntry ::=
  SEQUENCE {
    dot30amAdminState          INTEGER,
    dot30amOperStatus          INTEGER,
    dot30amMode                 INTEGER,
    dot30amMaxOamPduSize       Integer32,
    dot30amConfigRevision      Unsigned32,
    dot30amFunctionsSupported  BITS
  }
```

```
dot30amAdminState OBJECT-TYPE
  SYNTAX      INTEGER {
                disabled(1),
                enabled(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(1) state
    until an explicit management action or configuration
    information retained by the system causes a transition to the
    enabled(2) state."
```



Note that the value of this object is ignored when the interface is not operating in full-duplex mode. OAM is not supported on half-duplex links. "

REFERENCE "[[802.3ah](#)], 30.3.6.1.2"

```
::= { dot3OamEntry 1 }
```

dot3OamOperStatus OBJECT-TYPE

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

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 OAM sublayer.

This value is always disabled(1) if OAM is disabled on this interface via the dot3OamAdminState.

If the link has detected a fault and is transmitting OAMPDUS with a link fault indication, the value is linkFault(2).

The passiveWait(3) state is returned only by OAM entities in passive mode (dot3OamMode) and reflects the state in which

the

OAM entity is waiting to see if the peer device is OAM capable. The activeSendLocal(4) is used by active mode devices (dot3OamMode) 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 OAM 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

M. Squire

Expires - December 2005

[Page 12]

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 OAM PDU (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]. "

REFERENCE "[802.3ah], 30.3.6.1.4, 30.3.6.1.10, 30.3.6.1.11"  
 ::= { dot3OamEntry 2 }

dot3OamMode OBJECT-TYPE  
 SYNTAX INTEGER {  
     active(1),  
     passive(2)  
 }  
 MAX-ACCESS read-write  
 STATUS current  
 DESCRIPTION

in

"This object configures the mode of OAM operation for this Ethernet like interface. OAM on Ethernet interfaces may be

'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.

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). "

M. Squire

Expires - December 2005

[Page 13]

REFERENCE "[802.3ah], 30.3.6.1.3"  
 ::= { dot3OamEntry 3 }

dot3OamMaxOamPduSize OBJECT-TYPE

SYNTAX Integer32 (64..1522)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The largest OAMPDU that the OAM entity supports. OAM 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"

::= { dot3OamEntry 4 }

dot3OamConfigRevision OBJECT-TYPE

SYNTAX Unsigned32

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 configuration changes

have

occured which might require the peer OAM entity to re-

evaluate

whether the peering is allowed. See local\_satisfied in

[802.3ah, 57.3.1.2]. "

REFERENCE "[802.3ah], 30.3.6.1.12"

::= { dot3OamEntry 5 }

dot3OamFunctionsSupported 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 which 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. "

REFERENCE "[802.3ah], 30.3.6.1.6"

::= { dot3OamEntry 6 }



```

-----
--
-- Ethernet OAM Peer group
--

dot30amPeerTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Dot30amPeerEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Information about the OAM peer for a particular Ethernet
like
        interface. OAM entities communicate with a single OAM peer
        entity on full-duplex Ethernet links on which OAM is enabled
        and operating properly.

        In certain states, the OAM peer information is not available.
        Whether peer information is available is communicated via the
        dot30amPeerStatus object. When this object is inactive, all
        other information in the row is to be considered invalid. "
        ::= { dot30amMIB 2 }

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

        Note that there is at most one OAM peer for each Ethernet
like
        interface. There is exactly one row in this table for each
        Ethernet like interface supporting OAM. "
    INDEX      { ifIndex }
    ::= { dot30amPeerTable 1 }

Dot30amPeerEntry ::=
    SEQUENCE {
        dot30amPeerStatus          INTEGER,
        dot30amPeerMacAddress      MacAddress,
        dot30amPeerVendorOui       Dot30oui,
        dot30amPeerVendorInfo      Unsigned32,
        dot30amPeerMode            INTEGER,
        dot30amPeerMaxOamPduSize   Integer32,
        dot30amPeerConfigRevision  Unsigned32,
        dot30amPeerFunctionsSupported BITS
    }

```





## dot30amPeerStatus OBJECT-TYPE

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

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"This object indicates whether the information in this row should be considered valid. When active(1), the information is valid and represents the current peer of the OAM entity. When inactive(2), the information in this row is invalid.

A value of inactive(2) is returned if the dot30amOperStatus is disabled, passiveWait, or activeSendLocal. For all other values of dot30amOperStatus, a value of active(1) is returned.

"

REFERENCE "N/A"

::= { dot30amPeerEntry 1 }

## 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. This value

is initialized to all zeros (0x000000000000). This value is invalid if the dot30amPeerStatus is inactive.

An OAMPDU is indicated by a valid frame with (1) destination MAC address equal to that of the reserved MAC address for

Slow Protocols (See 43B of [\[802.3ah\]](#)), (2) a lengthOrType field equal to the reserved type for Slow Protocols, (3) and a Slow Protocols subtype equal to that of the subtype reserved for OAM. "

REFERENCE "[\[802.3ah\]](#), 30.3.6.1.5."

::= { dot30amPeerEntry 2 }

## dot30amPeerVendorOui OBJECT-TYPE

SYNTAX Dot30oui

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 OAM

M. Squire

Expires - December 2005

[Page 16]

entity. This value is initialized to all zeros (0x000000). This value is considered invalid if the dot30amPeerStatus is inactive.

An Information OAMPDU is indicated by a valid frame with (1) destination MAC address equal to that of the reserved MAC address for Slow Protocols (See 43B of [802.3ah]), (2) a lengthOrType field equal to the reserved type for Slow Protocols, (3) a Slow Protocols subtype equal to that of the subtype reserved for OAM, (4) a OAM code that equals the code reserved for Information OAMPDUs. "

REFERENCE "[802.3ah], 30.3.6.1.16."  
 ::= { dot30amPeerEntry 3 }

#### 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

vendor information field is within the Local Information TLV, and can be used to determine additional information about the peer entity. The format of the vendor information is unspecified within the 32-bit field. This value is

initialized

to all zeros (0x00000000). This value is invalid if the dot30amPeerStatus is inactive.

An Information OAMPDU is indicated by a valid frame with (1) destination MAC address equal to that of the reserved MAC address for Slow Protocols (See 43B of [802.3ah]), (2) a lengthOrType field equal to the reserved type for Slow Protocols, (3) a Slow Protocols subtype equal to that of the subtype reserved for OAM, and (4) a OAM code that equals the code reserved for Information OAMPDUs. "

REFERENCE "[802.3ah], 30.3.6.1.17."  
 ::= { dot30amPeerEntry 4 }

#### dot30amPeerMode OBJECT-TYPE

SYNTAX INTEGER {  
     active(1),  
     passive(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. This value is initialized to unknown(3), and is not valid if the dot30amPeerStatus is inactive.

An Information OAMPDU is indicated by a valid frame with (1) destination MAC address equal to that of the reserved MAC address for Slow Protocols (See 43B of [802.3ah]), (2) a lengthOrType field equal to the reserved type for Slow Protocols, (3) a Slow Protocols subtype equal to that of the subtype reserved for OAM, and (4) a OAM code that equals the code reserved for Information OAMPDUs. "

REFERENCE "[802.3ah], 30.3.6.1.7."

::= { dot30amPeerEntry 5 }

dot30amPeerMaxOamPduSize OBJECT-TYPE

SYNTAX Integer32 (64..1522)

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. This value is initialized to 64, and is invalid if the dot30amPeerStatus is inactive.

An Information OAMPDU is indicated by a valid frame with (1) destination MAC address equal to that of the reserved MAC address for Slow Protocols (See 43B of [802.3ah]), (2) a lengthOrType field equal to the reserved type for Slow Protocols, (3) a Slow Protocols subtype equal to that of the subtype reserved for OAM, and (4) a OAM code that equals the code reserved for Information OAMPDUs. "

REFERENCE "[802.3ah], 30.3.6.1.9."

::= { dot30amPeerEntry 6 }

dot30amPeerConfigRevision OBJECT-TYPE

SYNTAX Unsigned32

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 OAM this interface. This value is initialized to all zeros (0x00000000), and is invalid if the dot3OamPeerStatus is inactive.

Revision The configuration revision can be determined from the field of the Local Information TLV of the most recently received Information OAMPDU with a Local Information TLV.

An Information OAMPDU is indicated by a valid frame with (1) destination MAC address equal to that of the reserved MAC address for Slow Protocols (See 43B of [802.3ah]), (2) a lengthOrType field equal to the reserved type for Slow Protocols, (3) a Slow Protocols subtype equal to that of the subtype reserved for OAM, and (4) a OAM code that equals the code reserved for Information OAMPDU. "

REFERENCE "[802.3ah], 30.3.6.1.13."

::= { dot3OamPeerEntry 7 }

dot3OamPeerFunctionsSupported 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, and is invalid if the dot3OamPeerStatus is inactive.

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.

An Information OAMPDU is indicated by a valid frame with (1) destination MAC address equal to that of the reserved MAC address for Slow Protocols (See 43B of [802.3ah]), (2) a lengthOrType field equal to the reserved type for Slow Protocols, (3) a Slow Protocols subtype equal to that of the subtype reserved for OAM, and (4) a OAM code that equals the

```

code reserved for Information OAMPDUs. "
REFERENCE "[802.3ah], REFERENCE 30.3.6.1.7."
 ::= { dot30amPeerEntry 8 }

```

```

-----
--
--
--

```

```

dot30amLoopbackTable OBJECT-TYPE

```

```

SYNTAX SEQUENCE OF Dot30amLoopbackEntry

```

```

MAX-ACCESS not-accessible

```

```

STATUS current

```

```

DESCRIPTION

```

```

"This table contains methods to control the loopback state of
the local link as well as indicating the status of the
loopback function.

```

```

Loopback can be used to place the remote OAM entity in a

```

```

state

```

```

where every received frame (except OAMPDUs) are 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. "

```

```

 ::= { dot30amMIB 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. There is an
entry in this table for every Ethernet like interface on

```

```

which

```

```

supports OAM and loopback function within OAM (as indicated

```

```

in

```

```

dot30amFunctionsSupported). "

```

```

INDEX { ifIndex }

```

```

 ::= { dot30amLoopbackTable 1 }

```

```

Dot30amLoopbackEntry ::=

```

```

SEQUENCE {

```

```

dot30amLoopbackCommand INTEGER,

```

```

dot30amLoopbackStatus INTEGER,

```

```

dot30amLoopbackIgnoreRx INTEGER

```



}

M. Squire

Expires - December 2005

[Page 20]

## dot30amLoopbackCommand OBJECT-TYPE

```
SYNTAX      INTEGER {
                noLoopback (1),
                startRemoteLoopback (2),
                stopRemoteLoopback (3)
            }
```

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

"This attribute initiates or terminates remote loopback with an OAM peer. Writing startRemoteLoopback(2) to this

attribute

cause the local OAM client to send a loopback OAMPDU to the OAM peer with the loopback enable flags set. Writing stopRemoteLoopback(3) to this attribute will cause the local OAM client to send a loopback OAMPDU to the OAM peer with the loopback enable flags cleared. Writing noLoopback to this attribute has no effect.

Writes to this attribute are ignored unless the OAM status of this interface is 'operational' (dot30amOperStatus).

The attribute always returns noLoopback on a read. To determine the loopback status, use the attribute dot30amLoopbackStatus. "

REFERENCE "[[802.3ah](#)], 57.2.11"

::= { dot30amLoopbackEntry 1 }

## dot30amLoopbackStatus OBJECT-TYPE

```
SYNTAX      INTEGER {
                noLoopback (1),
                initiatingLoopback (2),
                remoteLoopback (3),
                terminatingLoopback (4),
                localLoopback (5),
                unknown (6)
            }
```

MAX-ACCESS read-only

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).

If the OAM client has sent an 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 acknowledgement that the remote OAM entity has received its loopback command request.

in If the local OAM client knows that the remote OAM entity is 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 The unknown(6) status indicates the parser and multiplexer combination is unexpected. This status may be returned if 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, 30.3.6.1.14, 30.3.6.1.15"  
 ::= { dot30amLoopbackEntry 2 }

dot30amLoopbackIgnoreRx OBJECT-TYPE  
 SYNTAX INTEGER { ignore(1), process(2) }  
 MAX-ACCESS read-write  
 STATUS current  
 DESCRIPTION

provide "Since OAM loopback is a disruptive operation (user traffic does not pass), this attribute provides a mechanism to 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)).

The attribute has no meaning if the local OAM entity does not support the loopback function (as defined in



```

dot30amFunctionsSupported). "
REFERENCE "[802.3ah], REFERENCE 57.2.11, 30.3.61.14,
30.3.6.1.15"
 ::= { dot30amLoopbackEntry 3 }

```

```

-----
--
-- Ethernet OAM Statistics group
--

```

```

dot30amStatsTable OBJECT-TYPE
SYNTAX      SEQUENCE OF Dot30amStatsEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"Statistics for the OAM function on a particular Ethernet
like
interface."
 ::= { dot30amMIB 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."
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
MAX-ACCESS  read-only
STATUS      current

```

DESCRIPTION

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

An Information OAMPDU is indicated by a valid frame with (1) destination MAC address equal to that of the reserved MAC address for Slow Protocols (See 43B of [[802.3ah](#)]), (2) a lengthOrType field equal to the reserved type for Slow Protocols, (3) a Slow Protocols subtype equal to that of the subtype reserved for OAM, and (4) an OAMPDU code equals the OAM Information 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](#)], 30.3.6.1.20."

::= { dot30amStatsEntry 1 }

dot30amInformationRx OBJECT-TYPE

```

SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current

```

DESCRIPTION

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

An Information OAMPDU is indicated by a valid frame with (1) destination MAC address equal to that of the reserved MAC address for Slow Protocols (See 43B of [[802.3ah](#)]), (2) a lengthOrType field equal to the reserved type for Slow Protocols, (3) a Slow Protocols subtype equal to that of the subtype reserved for OAM, and (4) an OAMPDU code equals the OAM Information 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](#)], 30.3.6.1.21."

::= { dot30amStatsEntry 2 }





dot30amUniqueEventNotificationTx OBJECT-TYPE

SYNTAX Counter32

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.

An Event Notification OAMPDU is indicated by a valid frame with (1) destination MAC address equal to that of the

reserved

MAC address for Slow Protocols (See 43B of [[802.3ah](#)]), (2) a lengthOrType field equal to the reserved type for Slow Protocols, (3) a Slow Protocols subtype equal to that of the subtype reserved for OAM, and (4) an OAMPDU code equals the OAM Event code.

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

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.

An Event Notification OAMPDU is indicated by a valid frame with (1) destination MAC address equal to that of the

reserved

MAC address for Slow Protocols (See 43B of [[802.3ah](#)]), (2) a lengthOrType field equal to the reserved type for Slow Protocols, (3) a Slow Protocols subtype equal to that of the subtype reserved for OAM, and (4) an OAMPDU code equals the OAM Event code.



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

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.

An Event Notification OAMPDU is indicated by a valid frame with (1) destination MAC address equal to that of the reserved MAC address for Slow Protocols (See 43B of [[802.3ah](#)]), (2) a lengthOrType field equal to the reserved type for Slow Protocols, (3) a Slow Protocols subtype equal to that of the subtype reserved for OAM, and (4) an OAMPDU code equals the OAM Event code.

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

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.

reserved An Event Notification OAMPDU is indicated by a valid frame with (1) destination MAC address equal to that of the MAC address for Slow Protocols (See 43B of [[802.3ah](#)]), (2) a lengthOrType field equal to the reserved type for Slow Protocols, (3) a Slow Protocols subtype equal to that of the subtype reserved for OAM, and (4) an OAMPDU code equals the OAM Event code.

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

initialization 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."  
 ::= { dot3OamStatsEntry 6 }

dot3OamLoopbackControlTx OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

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

An Loopback Control OAMPDU is indicated by a valid frame with (1) destination MAC address equal to that of the reserved MAC address for Slow Protocols (See 43B of [[802.3ah](#)]), (2) a lengthOrType field equal to the reserved type for Slow Protocols, (3) a Slow Protocols subtype equal to that of the subtype reserved for OAM, and (4) an OAMPDU code equals the OAM Loopback Control code.

initialization 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."  
 ::= { dot3OamStatsEntry 7 }

dot3OamLoopbackControlRx OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current  
DESCRIPTION

M. Squire

Expires - December 2005

[Page 27]

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

An Loopback Control OAMPDU is indicated by a valid frame with (1) destination MAC address equal to that of the reserved MAC address for Slow Protocols (See 43B of [802.3ah]), (2) a lengthOrType field equal to the reserved type for Slow Protocols, (3) a Slow Protocols subtype equal to that of the subtype reserved for OAM, and (4) an OAMPDU code equals the OAM Loopback Control 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], 30.3.6.1.27."  
 ::= { dot3OamStatsEntry 8 }

dot3OamVariableRequestTx OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

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

An Variable Request OAMPDU is indicated by a valid frame with (1) destination MAC address equal to that of the reserved MAC address for Slow Protocols (See 43B of [802.3ah]), (2) a lengthOrType field equal to the reserved type for Slow Protocols, (3) a Slow Protocols subtype equal to that of the subtype reserved for OAM, and (4) an OAMPDU code equals the OAM Variable Request 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], 30.3.6.1.28."  
 ::= { dot3OamStatsEntry 9 }

dot3OamVariableRequestRx OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

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

An Variable Request OAMPDU is indicated by a valid frame with

(1) destination MAC address equal to that of the reserved MAC

M. Squire

Expires - December 2005

[Page 28]



address for Slow Protocols (See 43B of [\[802.3ah\]](#)), (2) a lengthOrType field equal to the reserved type for Slow Protocols, (3) a Slow Protocols subtype equal to that of the subtype reserved for OAM, and (4) an OAMPDU code equals the OAM Variable Request 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 ["\[802.3ah\]](#), 30.3.6.1.29."

::= { dot3OamStatsEntry 10 }

dot3OamVariableResponseTx OBJECT-TYPE

SYNTAX Counter32  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION

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

An Variable Response OAMPDU is indicated by a valid frame

with

(1) destination MAC address equal to that of the reserved MAC address for Slow Protocols (See 43B of [\[802.3ah\]](#)), (2) a lengthOrType field equal to the reserved type for Slow Protocols, (3) a Slow Protocols subtype equal to that of the subtype reserved for OAM, and (4) an OAMPDU code equals the OAM Variable Response 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 ["\[802.3ah\]](#), 30.3.6.1.30."

::= { dot3OamStatsEntry 11 }

dot3OamVariableResponseRx OBJECT-TYPE

SYNTAX Counter32  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION

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

An Variable Response OAMPDU is indicated by a valid frame

with

(1) destination MAC address equal to that of the reserved MAC address for Slow Protocols (See 43B of [\[802.3ah\]](#)), (2) a lengthOrType field equal to the reserved type for Slow Protocols, (3) a Slow Protocols subtype equal to that of the subtype reserved for OAM, and (4) an OAMPDU code equals the OAM Variable Response 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](#)], 30.3.6.1.31."

::= { dot30amStatsEntry 12 }

## dot30amOrgSpecificTx OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

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

reserved

An Organization Specific OAMPDU is indicated by a valid frame with (1) destination MAC address equal to that of the MAC address for Slow Protocols (See 43B of [[802.3ah](#)]), (2) a lengthOrType field equal to the reserved type for Slow Protocols, (3) a Slow Protocols subtype equal to that of the subtype reserved for OAM, and (4) an OAMPDU code equals the OAM Organization Specific 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](#)], 30.3.6.1.32."

::= { dot30amStatsEntry 13 }

## dot30amOrgSpecificRx OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

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

reserved

An Organization Specific OAMPDU is indicated by a valid frame with (1) destination MAC address equal to that of the MAC address for Slow Protocols (See 43B of [[802.3ah](#)]), (2) a lengthOrType field equal to the reserved type for Slow Protocols, (3) a Slow Protocols subtype equal to that of the subtype reserved for OAM, and (4) an OAMPDU code equals the OAM Organization Specific 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. "

M. Squire

Expires - December 2005

[Page 30]

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

dot30amUnsupportedCodesTx OBJECT-TYPE

SYNTAX Counter32  
 MAX-ACCESS read-only  
 STATUS current

DESCRIPTION

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

An unsupported opcode OAMPDU is indicated by a valid frame with (1) destination MAC address equal to that of the

reserved

MAC address for Slow Protocols (See 43B of [802.3ah]), (2) a lengthOrType field equal to the reserved type for Slow Protocols, (3) a Slow Protocols subtype equal to that of the subtype reserved for OAM, and (4) an OAMPDU code equals the opcode for a function that is not supported by the device.

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  
 MAX-ACCESS read-only  
 STATUS current

DESCRIPTION

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

An unsupported opcode OAMPDU is indicated by a valid frame with (1) destination MAC address equal to that of the

reserved

MAC address for Slow Protocols (See 43B of [802.3ah]), (2) a lengthOrType field equal to the reserved type for Slow Protocols, (3) a Slow Protocols subtype equal to that of the subtype reserved for OAM, and (4) an OAMPDU code equals the opcode for a function that is not supported by the device.

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



MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A count of the number of frames that were dropped by the OAM 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. 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 notifications to indicate various link problems. This table contains the mechanisms to configure the thresholds to generate the standard Ethernet OAM events.

These 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 (e.g. 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 (e.g. 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 (e.g. 10 frames in 1 second had errors).

- Errored Frame Seconds Summary Event. Generated when the number of errored frame seconds exceeds a threshold

within





defined a given time period (e.g. 10 errored frame seconds within the last 100 seconds). An errored frame second is

```

as a 1 second interval which had >0 frame errors.
"
 ::= { dot30amMIB 5 }

```

dot30amEventConfigEntry OBJECT-TYPE

SYNTAX Dot30amEventConfigEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Event configuration information is available for every Ethernet like interface that supports OAM and the event function of OAM as indicated in the dot30amFunctionsSupported attribute.

Event configuration controls when the local management entity sends Event Notification OAMPDUs to its OAM peer. "

INDEX { ifIndex }

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

Dot30amEventConfigEntry ::=

SEQUENCE {

```

    dot30amErrSymPeriodWindowHi      Unsigned32,
    dot30amErrSymPeriodWindowLo      Unsigned32,
    dot30amErrSymPeriodThresholdHi   Unsigned32,
    dot30amErrSymPeriodThresholdLo   Unsigned32,
    dot30amErrSymPeriodEvNotifEnable INTEGER,
    dot30amErrFramePeriodWindow      Unsigned32,
    dot30amErrFramePeriodThreshold   Unsigned32,
    dot30amErrFramePeriodEvNotifEnable INTEGER,
    dot30amErrFrameWindow             Unsigned32,
    dot30amErrFrameThreshold          Unsigned32,
    dot30amErrFrameEvNotifEnable     INTEGER,
    dot30amErrFrameSecsSummaryWindow Integer32,
    dot30amErrFrameSecsSummaryThreshold Integer32,
    dot30amErrFrameSecsEvNotifEnable INTEGER

```

```
 }
```

dot30amErrSymPeriodWindowHi OBJECT-TYPE

SYNTAX Unsigned32

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





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 the threshold has been crossed in this window. "

REFERENCE "[[802.3ah](#)], 30.3.6.1.34"

::= { dot30amEventConfigEntry 3 }

dot30amErrSymPeriodThresholdLo OBJECT-TYPE

SYNTAX Unsigned32

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 the threshold has been crossed in this window. "

REFERENCE "[[802.3ah](#)], 30.3.6.1.34"

::= { dot30amEventConfigEntry 4 }

dot30amErrSymPeriodEvNotifEnable OBJECT-TYPE

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

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Indicates whether the occurrence of Errored Symbol Period Events should result in Event Notification OAMPDUs generated by the OAM layer.

By default, this object should have the value enabled(1) 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.  
"

REFERENCE "N/A"

::= { dot30amEventConfigEntry 5 }



## dot30amErrFramePeriodWindow OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

"The number of frames over which the threshold is defined.

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 the threshold has been crossed in this window. "

REFERENCE "[[802.3ah](#)], 30.3.6.1.38"

::= { dot30amEventConfigEntry 6 }

## dot30amErrFramePeriodThreshold OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

"The number of frame errors that must occur for this event to be triggered.

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 the threshold has been crossed in this window. "

REFERENCE "[[802.3ah](#)], 30.3.6.1.38"

::= { dot30amEventConfigEntry 7 }

## dot30amErrFramePeriodEvNotifEnable OBJECT-TYPE

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

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

"Indicates whether the occurrence of an Errored Frame Period Event should result in an Event Notification OAMPDU generated by the OAM layer.

By default, this object should have the value enabled(1) 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.

"

REFERENCE "N/A"

::= { dot30amEventConfigEntry 8 }

## dot30amErrFrameWindow OBJECT-TYPE

SYNTAX Unsigned32  
 MAX-ACCESS read-write  
 STATUS current  
 DESCRIPTION

"The amount of time (in 100ms increments) over which the threshold is defined.

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 the threshold has been crossed in this window. "

REFERENCE "[[802.3ah](#)], 30.3.6.1.36"  
 ::= { dot30amEventConfigEntry 9 }

dot30amErrFrameThreshold OBJECT-TYPE

SYNTAX Unsigned32  
 MAX-ACCESS read-write  
 STATUS current  
 DESCRIPTION

"The number of frame errors that must occur for this event to be triggered.

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"  
 ::= { dot30amEventConfigEntry 10 }

dot30amErrFrameEvNotifEnable OBJECT-TYPE

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

"Indicates whether the occurrence of an Errored Frame Event should result in an Event Notification OAMPDU generated by

the

OAM layer.

By default, this object should have the value enabled(1) 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.

"

REFERENCE "N/A"  
 ::= { dot30amEventConfigEntry 11 }

dot30amErrFrameSecsSummaryWindow OBJECT-TYPE





SYNTAX Integer32 (100..9000)

MAX-ACCESS read-write

STATUS current

DESCRIPTION

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

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

REFERENCE "[[802.3ah](#)], 30.3.6.1.40"  
 ::= { dot30amEventConfigEntry 12 }

dot30amErrFrameSecsSummaryThreshold OBJECT-TYPE

SYNTAX Integer32 (1..900)

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The number of errored frame seconds that must occur for this event to be triggered.

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

REFERENCE "[[802.3ah](#)], 30.3.6.1.40"  
 ::= { dot30amEventConfigEntry 13 }

dot30amErrFrameSecsEvNotifEnable OBJECT-TYPE

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

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Indicates whether the occurrence of an Errored Frame Seconds Summary Event should result in an Event Notification OAMPDU generated by the OAM layer.

By default, this object should have the value enabled(1) 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."

REFERENCE "N/A"

::= { dot30amEventConfigEntry 14 }

---

M. Squire

Expires - December 2005

[Page 38]

```
--
--
--
```

```
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.  "
    ::= { dot30amMIB 6 }
```

```
dot30amEventLogEntry OBJECT-TYPE
    SYNTAX      Dot30amEventLogEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "An entry in the dot30amEventLogTable."
    INDEX       { ifIndex, dot30amEventLogIndex }
    ::= { dot30amEventLogTable 1 }
```

```
Dot30amEventLogEntry ::=
    SEQUENCE {
        dot30amEventLogIndex          Unsigned32,
        dot30amEventLogTimestamp      DateAndTime,
        dot30amEventLogOui             Dot30Oui,
        dot30amEventLogType            Unsigned32,
        dot30amEventLogLocation        INTEGER,
        dot30amEventLogWindowHi        Unsigned32,
        dot30amEventLogWindowLo        Unsigned32,
        dot30amEventLogThresholdHi     Unsigned32,
        dot30amEventLogThresholdLo     Unsigned32,
        dot30amEventLogValue           CounterBasedGauge64,
        dot30amEventLogRunningTotal    CounterBasedGauge64,
        dot30amEventLogEventTotal      Unsigned32
    }
```

```
dot30amEventLogIndex OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An arbitrary integer for identifying individual events
        within the event log.  "
```

```
REFERENCE "N/A"
 ::= { dot30amEventLogEntry 1 }
```

```
dot30amEventLogTimestamp OBJECT-TYPE
    SYNTAX      DateAndTime
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The date and time that this event instance occurred. "
    REFERENCE   "N/A"
    ::= { dot30amEventLogEntry 2 }
```

```
dot30amEventLogOui OBJECT-TYPE
    SYNTAX      Dot3oui
    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 which
        gets reflected here. "
```

```
REFERENCE "N/A"
 ::= { 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 0x0180C2, 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 Fs (0xFFFFFFFF).  
"

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, or was received from

the

OAM peer via Ethernet OAM."

REFERENCE "N/A"

::= { 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 (e.g. 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 (0xFFFFFFFF) and adds no useful information. "

REFERENCE "[[802.3ah](#)], 30.3.6.1.37 and 57.5.3.2."

::= { 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 (e.g. 5, when 11 occurrences happened in 5 seconds while the threshold was 10). The two objects are combined as:

```
dot30amEventLogWindow = ((2^32) * dot30amEventLogWindowHi)
                        + dot30amEventLogWindowLo
```

Otherwise, this value is returned as all F's (0xFFFFFFFF) and adds no useful information. "

REFERENCE "[[802.3ah](#)], 30.3.6.1.37 and 57.5.3.2."  
 ::= { 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 (e.g. 10, when 11 occurrences happened in 5 seconds while the threshold was

10).

The two objects are combined as:

```
dot30amEventLogThreshold = ((2^32) *
dot30amEventLogThresholdHi)
                        + dot30amEventLogThresholdLo
```

Otherwise, this value is returned as all F's (0xFFFFFFFF) 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 (e.g. 10, when 11 occurrences happened in 5 seconds while the threshold was

10).

The two objects are combined as:

```
dot30amEventLogThreshold = ((2^32) *
dot30amEventLogThresholdHi)
                        + dot30amEventLogThresholdLo
```

Otherwise, this value is returned as all F's (0xFFFFFFFF) 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 (e.g. 11, when 11
occurrences
        happened in 5 seconds while the threshold was 10).

        Otherwise, this value is returned as all F's
        (0xFFFFFFFFFFFFFFFF) 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
        "This value represents the total number of times this
        occurrence has happened since the last reset (e.g. 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
        "This value represents the total number of times one or ore
        of these occurrences have resulted in an event (e.g. 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
--

dot30amTraps OBJECT IDENTIFIER ::= { dot30amMIB 7 }

```



dot30amTrapsPrefix OBJECT IDENTIFIER ::= {dot30amTraps 0}

dot30amThresholdEvent NOTIFICATION-TYPE

```

OBJECTS { ifIndex,
          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 indicating a threshold event.

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

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

::= { dot30amTrapsPrefix 1 }

dot30amNonThresholdEvent NOTIFICATION-TYPE

```

OBJECTS { ifIndex,
          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 indicating a non-threshold crossing

event.



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

The management entity should periodically check dot30amEventLogTable to detect any missed events."  
 ::= { dot30amTrapsPrefix 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 OAM
    implementations that support loopback functionality. "

GROUP             dot30amErrSymbolPeriodEventGroup
DESCRIPTION
    "This group is mandatory for all IEEE 802.3 OAM
    implementations that support event functionality. "

GROUP             dot30amErrFramePeriodEventGroup
DESCRIPTION
    "This group is mandatory for all IEEE 802.3 OAM
    implementations that support event functionality. "

GROUP             dot30amErrFrameEventGroup
DESCRIPTION
    "This group is mandatory for all IEEE 802.3 OAM
    implementations that support event functionality. "
```

```

GROUP      dot30amErrFrameSecsSummaryEventGroup
DESCRIPTION
  "This group is mandatory for all IEEE 802.3 OAM
  implementations that support event functionality. "

GROUP      dot30amEventLogGroup
DESCRIPTION
  "This group is optional for all IEEE 802.3 OAM
  implementations. "

GROUP      dot30amNotificationGroup
DESCRIPTION
  "This group is optional for all IEEE 802.3 OAM
  implementations. "

 ::= { 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      { dot30amPeerStatus,
               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      {  dot30amLoopbackCommand,
                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 }
```

```
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 9 }
```

```
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
```

```
occured
```

```
  on a specified Ethernet link."
```

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

```
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 authorization 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

M. Squire

Expires - December 2005

[Page 49]

disabled on Ethernet-like 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 operation of OAM 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 only affect the operation of the OAM protocol itself and not user data traffic.

The loopback capability of OAM can have potentially disruptive effects in that the 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.

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. References

### 8.1 Normative References

[802.3ah] Institute of Electrical and Electronic Engineers, IEEE Draft Std 802.3ah-2004-2002 Draft 3.3, "IEEE Standard for Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications - Draft amendment 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", May 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.

[802-2001] Institute of Electrical and Electronic Engineers, IEEE Std 802-2001, "Standard for Local and Metropolitan Area Networks: Architecture and Overview", March 2002.

[RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", [BCP 9](#), [RFC 2026](#), October 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997

[RFC2578] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, [RFC 2578](#), April 1999.

[RFC2579] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "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.





[RFC3668] Bradner, S. "Intellectual Property Rights in IETF Technology", [BCP 79](#), [RFC 3668](#), February 2004.

## **8.2 Informative References**

[802.3ah-copper] Beili, Ed, "Managed Objects for Ethernet Passive Optical Networks", [draft-ietf-hubmib-efm-epon-mib-021.txt](#), April October 2004.

[802.3ah-epon] Khermosh, Lior, "Ethernet in the First Mile Copper (EFMCu) Interfaces MIB", [draft-ietf-hubmib-efm-cu-mib-020.txt](#), January October 2004.

[RFC2665] Flick, J. and Johnson J. "Definitions of Managed Objects for the Ethernet-likeEthernet like Interface Types", STD 58, [RFC 2580](#), April 1999.

[RFC2863] McCloghrie, K., Kastenholz, F., "The Interfaces Group MIB", [RFC 2863](#), June 2000.

[RFC3410] Case, J., Mundy, R., Partain, D., Stewart, B., "Introduction and Applicability Statements for Internet Standard Management Framework", [RFC 3410](#), December 2003.

[RFC3635] Flick, J., "Definitions of Managed Objects for the Ethernet-likeEthernet like Interface Types", [RFC 3635](#), September 2003.

[RFC3636] Flick, J., "Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs)", [RFC 3636](#), September 2003.

## **9. 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 (Chair, IETF HUBMIB WG)  
Jonathan Thatcher  
Geoff Thompson

## **10. Author's Address**

Note: Author's email address is spelled out to help protect against email address harvesting programs.

Matt Squire  
Hatteras Networks  
639 Davis Drive, Suite 200  
Phone: +1-919-991-5460  
Email: msquire at hatterasnetworks dot com

## **11. Intellectual Property Statement**

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

M. Squire

Expires - December 2005

[Page 53]

this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

The IETF has been notified of intellectual property rights claimed in regard to some or all of the specification contained in this document. For more information consult the online list of claimed rights.

## **12. Copyright Statement**

Copyright (C) The Internet Society (2004). 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 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.

## Acknowledgement

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

