

Network Working Group
Request for Comments: 2665
Obsoletes: 2358
Category: Standards Track

J. Flick
Hewlett-Packard Company
J. Johnson
RedBack Networks
August 1999

Definitions of Managed Objects for
the Ethernet-like Interface Types

Status of this Memo

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

Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. This memo obsoletes RFC 2358, "Definitions of Managed Objects for the Ethernet-like Interface Types". This memo extends that specification by including management information useful for the management of 1000 Mb/s and full-duplex Ethernet interfaces.

Ethernet technology, as defined by the 802.3 Working Group of the IEEE, continues to evolve, with scalable increases in speed, new types of cabling and interfaces, and new features. This evolution may require changes in the managed objects in order to reflect this new functionality. This document, as with other documents issued by this working group, reflects a certain stage in the evolution of Ethernet technology. In the future, this document might be revised, or new documents might be issued by the Ethernet Interfaces and Hub MIB Working Group, in order to reflect the evolution of Ethernet technology.

Table of Contents

1. Introduction	2
2. The SNMP Management Framework	3
3. Overview	4
3.1. Relation to MIB-2	4
3.2. Relation to the Interfaces MIB	5
3.2.1. Layering Model	5
3.2.2. Virtual Circuits	5
3.2.3. ifTestTable	5
3.2.4. ifRcvAddressTable	6
3.2.5. ifPhysAddress	6
3.2.6. ifType	6
3.2.7. Specific Interface MIB Objects	7
3.3. Relation to the 802.3 MAU MIB	11
3.4. dot3StatsEtherChipSet	11
3.5. Mapping of IEEE 802.3 Managed Objects	12
4. Definitions	16
5. Intellectual Property	39
6. Acknowledgements	40
7. References	41
8. Security Considerations	43
9. Authors' Addresses	44
A. Change Log	45
A.1. Changes since RFC 2358	45
A.2. Changes between RFC 1650 and RFC 2358	46
B. Full Copyright Statement	47

1. Introduction

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it defines objects for managing Ethernet-like interfaces.

This memo also includes a MIB module. This MIB module extends the list of managed objects specified in the earlier version of this MIB: RFC 2358 [23].

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 [26].

2. The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

- o An overall architecture, described in RFC 2571 [1].
- o Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIV1 and described in STD 16, RFC 1155 [2], STD 16, RFC 1212 [3] and RFC 1215 [4]. The second version, called SMIV2, is described in STD 58, RFC 2578 [5], STD 58, RFC 2579 [6] and STD 58, RFC 2580 [7].
- o Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPV1 and described in STD 15, RFC 1157 [8]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPV2C and described in RFC 1901 [9] and RFC 1906 [10]. The third version of the message protocol is called SNMPV3 and described in RFC 1906 [10], RFC 2572 [11] and RFC 2574 [12].
- o Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [8]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [13].
- o A set of fundamental applications described in RFC 2573 [14] and the view-based access control mechanism described in RFC 2575 [15].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms defined in the SMI.

This memo specifies a MIB module that is compliant to the SMIV2. A MIB conforming to the SMIV1 can be produced through the appropriate translations. The resulting translated MIB must be semantically equivalent, except where objects or events are omitted because no translation is possible (use of Counter64). Some machine readable information in SMIV2 will be converted into textual descriptions in SMIV1 during the translation process. However, this loss of machine readable information is not considered to change the semantics of the MIB.

3. Overview

Instances of these object types represent attributes of an interface to an ethernet-like communications medium. At present, ethernet-like media are identified by the following values of the ifType object in the Interfaces MIB [25]:

```
ethernetCsmacd(6)
iso88023Csmacd(7)
starLan(11)
```

The definitions presented here are based on Section 30, "10 Mb/s, 100 Mb/s and 1000 Mb/s Management", and Annex 30A, "GDMO Specification for 802.3 managed object classes" of IEEE Std. 802.3, 1998 Edition [16], as originally interpreted by Frank Kastenholz then of Interlan in [17]. Implementors of these MIB objects should note that IEEE Std. 802.3 [16] explicitly describes (in the form of Pascal pseudocode) when, where, and how various MAC attributes are measured. The IEEE document also describes the effects of MAC actions that may be invoked by manipulating instances of the MIB objects defined here.

To the extent that some of the attributes defined in [16] are represented by previously defined objects in MIB-2 [24] or in the Interfaces MIB [25], such attributes are not redundantly represented by objects defined in this memo. Among the attributes represented by objects defined in other memos are the number of octets transmitted or received on a particular interface, the number of frames transmitted or received on a particular interface, the promiscuous status of an interface, the MAC address of an interface, and multicast information associated with an interface.

3.1. Relation to MIB-2

This section applies only when this MIB is used in conjunction with the "old" (RFC 1213) [24] interface group.

The relationship between an ethernet-like interface and an interface in the context of MIB-2 is one-to-one. As such, the value of an ifIndex object instance can be directly used to identify corresponding instances of the objects defined herein.

For agents which implement the (now deprecated) ifSpecific object, an instance of that object that is associated with an ethernet-like interface has the OBJECT IDENTIFIER value:

```
dot3    OBJECT IDENTIFIER ::= { transmission 7 }
```

3.2. Relation to the Interfaces MIB

The Interface MIB [25] requires that any MIB which is an adjunct of the Interface MIB clarify specific areas within the Interface MIB. These areas were intentionally left vague in the Interface MIB to avoid over constraining the MIB, thereby precluding management of certain media-types.

Section 3.3 of [25] enumerates several areas which a media-specific MIB must clarify. Each of these areas is addressed in a following subsection. The implementor is referred to [25] in order to understand the general intent of these areas.

3.2.1. Layering Model

This MIB does not provide for layering. There are no sublayers.

EDITOR'S NOTE:

One could foresee the development of an 802.2 and enet-transceiver MIB. They could be higher and lower sublayers, respectively. All that THIS document should do is allude to the possibilities and urge the implementor to be aware of the possibility and that they may have requirements which supersede the requirements in this document.

3.2.2. Virtual Circuits

This medium does not support virtual circuits and this area is not applicable to this MIB.

3.2.3. ifTestTable

This MIB defines two tests for media which are instrumented with this MIB; TDR and Loopback. Implementation of these tests is not required. Many common interface chips do not support one or both of these tests.

These two tests are provided as a convenience, allowing a common method to invoke the test.

Standard MIBs do not include objects in which to return the results of the TDR test. Any needed objects MUST be provided in the vendor specific MIB.

Note that the ifTestTable is now deprecated. Work is underway to define a replacement MIB for system and interface testing. It is expected that the tests defined in this document will be usable in this replacement MIB.

3.2.4. ifRcvAddressTable

This table contains all IEEE 802.3 addresses, unicast, multicast, and broadcast, for which this interface will receive packets and forward them up to a higher layer entity for local consumption. The format of the address, contained in ifRcvAddressAddress, is the same as for ifPhysAddress.

In the event that the interface is part of a MAC bridge, this table does not include unicast addresses which are accepted for possible forwarding out some other port. This table is explicitly not intended to provide a bridge address filtering mechanism.

3.2.5. ifPhysAddress

This object contains the IEEE 802.3 address which is placed in the source-address field of any Ethernet, Starlan, or IEEE 802.3 frames that originate at this interface. Usually this will be kept in ROM on the interface hardware. Some systems may set this address via software.

In a system where there are several such addresses the designer has a tougher choice. The address chosen should be the one most likely to be of use to network management (e.g. the address placed in ARP responses for systems which are primarily IP systems).

If the designer truly can not chose, use of the factory- provided ROM address is suggested.

If the address can not be determined, an octet string of zero length should be returned.

The address is stored in binary in this object. The address is stored in "canonical" bit order, that is, the Group Bit is positioned as the low-order bit of the first octet. Thus, the first byte of a multicast address would have the bit 0x01 set.

3.2.6. ifType

This MIB applies to interfaces which have any of the following ifType values:

```
ethernetCsmacd(6)
iso88023Csmacd(7)
starLan(11)
```

It is RECOMMENDED that all Ethernet-like interfaces use an ifType of ethernetCsmacd(6) regardless of the speed that the interface is running or the link-layer encapsulation in use. iso88023Csmacd(7) and starLan(11) are supported for backwards compatibility.

There are three other interface types defined in the IANAifType-MIB for Ethernet. They are fastEther(62), fastEtherFX(69), and gigabitEthernet(117). This document takes the position that an Ethernet is an Ethernet, and Ethernet interfaces SHOULD always have the same value of ifType. Information on the particular flavor of Ethernet that an interface is running is available from ifSpeed in the Interfaces MIB, and ifMauType in the 802.3 MAU MIB. An Ethernet-like interface SHOULD NOT use the fastEther(62), fastEtherFX(69), or gigabitEthernet(117) ifTypes.

Interfaces with any of the supported ifType values map to the EtherLike-MIB in the same manner. There are no implementation differences.

3.2.7. Specific Interface MIB Objects

The following table provides specific implementation guidelines for applying the interface group objects to ethernet-like media.

Object	Guidelines
ifIndex	Each ethernet-like interface is represented by an ifEntry. The dot3StatsTable in this MIB module is indexed by dot3StatsIndex. The interface identified by a particular value of dot3StatsIndex is the same interface as identified by the same value of ifIndex.
ifDescr	Refer to [25].
ifType	Refer to section 3.2.6.
ifMtu	1500 octets. NOTE: This is the MTU as seen by the MAC client. When a higher layer protocol, like IP, is running over Ethernet, this is the MTU that will be seen by that higher layer protocol. However, when using the IEEE 802.2 LLC protocol, higher layer protocols will see a different MTU. In particular, an LLC type 1 client protocol will see

an MTU of 1497 octets, and a protocol running over SNAP will see an MTU of 1492 octets.

ifSpeed	The current operational speed of the interface in bits per second. For current ethernet-like interfaces, this will be equal to 1,000,000 (1 million), 10,000,000 (10 million), 100,000,000 (100 million), or 1,000,000,000 (1 billion). If the interface implements auto-negotiation, auto-negotiation is enabled for this interface, and the interface has not yet negotiated to an operational speed, this object SHOULD reflect the maximum speed supported by the interface. Note that this object MUST NOT indicate a doubled value when operating in full-duplex mode. It MUST indicate the correct line speed regardless of the current duplex mode. The duplex mode of the interface may be determined by examining either the dot3StatsDuplexStatus object in this MIBmodule, or the ifMauType object in the 802.3 MAU MIB.
ifPhysAddress	Refer to section 3.2.5.
ifAdminStatus	Write access is not required. Support for 'testing' is not required.
ifOperStatus	The operational state of the interface. Support for 'testing' is not required. The value 'dormant' has no meaning for an ethernet-like interface.
ifLastChange	Refer to [25].
ifInOctets	The number of octets in valid MAC frames received on this interface, including the MAC header and FCS. This does include the number of octets in valid MAC Control frames received on this interface.

ifInUcastPkts	Refer to [25]. Note that this does not include MAC Control frames, since MAC Control frames are consumed by the interface layer and are not passed to any higher layer protocol.
ifInDiscards	Refer to [25].
ifInErrors	The sum for this interface of dot3StatsAlignmentErrors, dot3StatsFCSErrors, dot3StatsFrameTooLongs, dot3StatsInternalMacReceiveErrors and dot3StatsSymbolErrors.
ifInUnknownProtos	Refer to [25].
ifOutOctets	The number of octets transmitted in valid MAC frames on this interface, including the MAC header and FCS. This does include the number of octets in valid MAC Control frames transmitted on this interface.
ifOutUcastPkts	Refer to [25]. Note that this does not include MAC Control frames, since MAC Control frames are generated by the interface layer, and are not passed from any higher layer protocol.
ifOutDiscards	Refer to [25].
ifOutErrors	The sum for this interface of: dot3StatsSQETestErrors, dot3StatsLateCollisions, dot3StatsExcessiveCollisions, dot3StatsInternalMacTransmitErrors and dot3StatsCarrierSenseErrors.
ifName	Locally-significant textual name for the interface (e.g. lan0).
ifInMulticastPkts	Refer to [25]. Note that this does not include MAC Control frames, since MAC Control frames are consumed by the interface layer and are not passed to any higher layer protocol.

ifInBroadcastPkts	Refer to [25]. Note that this does not include MAC Control frames, since MAC Control frames are generated by the interface layer, and are not passed from any higher layer protocol.
ifOutMulticastPkts	Refer to [25]. Note that this does not include MAC Control frames, since MAC Control frames are consumed by the interface layer and are not passed to any higher layer protocol.
ifOutBroadcastPkts	Refer to [25]. Note that this does not include MAC Control frames, since MAC Control frames are generated by the interface layer, and are not passed from any higher layer protocol.
ifHCInOctets ifHCOctets	64-bit versions of counters. Required for ethernet-like interfaces that are capable of operating at 20Mbit/sec or faster, even if the interface is currently operating at less than 20Mbit/sec.
ifHCInUcastPkts ifHCInMulticastPkts ifHCInBroadcastPkts ifHCOutUcastPkts ifHCOutMulticastPkts ifHCOutBroadcastPkts	64-bit versions of packet counters. Required for ethernet-like interfaces that are capable of operating at 640Mbit/sec or faster, even if the interface is currently operating at less than 640Mbit/sec.
ifLinkUpDownTrapEnable	Refer to [25]. Default is 'enabled'
ifHighSpeed	The current operational speed of the interface in millions of bits per second. For current ethernet-like interfaces, this will be equal to 1, 10, 100, or 1,000. If the interface implements auto-negotiation, auto-negotiation is enabled for this interface, and the interface has not yet negotiated to an operational speed, this object SHOULD reflect the maximum speed supported by the interface. Note that this object MUST NOT indicate a doubled value when operating in full-duplex mode. It MUST indicate the

correct line speed regardless of the current duplex mode. The duplex mode of the interface may be determined by examining either the dot3StatsDuplexStatus object in this MIB module, or the ifMauType object in the 802.3 MAU MIB.

ifPromiscuousMode	Refer to [25].
ifConnectorPresent	This will normally be 'true'.
ifAlias	Refer to [25].
ifCounterDiscontinuityTime	Refer to [25]. Note that a discontinuity in the Interface MIB counters may also indicate a discontinuity in some or all of the counters in this MIB that are associated with that interface.
ifStackHigherLayer	Refer to section 3.2.1.
ifStackLowerLayer	
ifStackStatus	
ifRcvAddressAddress	Refer to section 3.2.4.
ifRcvAddressStatus	
ifRcvAddressType	

3.3. Relation to the 802.3 MAU MIB

Support for the mauModIfCompl2 compliance statement of the MAU-MIB [27] is REQUIRED for Ethernet-like interfaces. This MIB is needed in order to allow applications to determine the current MAU type in use by the interface, and to control autonegotiation and duplex mode for the interface. Implementing this MIB module without implementing the MAU-MIB would leave applications with no standard way to determine the media type in use, and no standard way to control the duplex mode of the interface.

3.4. dot3StatsEtherChipSet

This document defines an object called dot3StatsEtherChipSet, which is used to identify the MAC hardware used to communicate on an interface. Previous versions of this document contained a number of OID assignments for some existing Ethernet chipsets. Maintaining

that list as part of this document has proven to be problematic, so the OID assignments contained in previous versions of this document have now been moved to a separate document [28].

The dot3StatsEtherChipSet object has now been deprecated. Implementation feedback indicates that this object is much more useful in theory than in practice. The object's utility in debugging network problems in the field appears to be limited. In those cases where it may be useful, it is not sufficient, since it identifies only the MAC chip, and not the PHY, PMD, or driver. The administrative overhead involved in maintaining a central registry of chipset OIDs cannot be justified for an object whose usefulness is questionable at best.

Implementations which continue to support this object for the purpose of backwards compatibility may continue to use the values defined in [28]. For chipsets not listed in [28], implementors should assign OBJECT IDENTIFIERS within that part of the registration tree delegated to individual enterprises.

3.5. Mapping of IEEE 802.3 Managed Objects

IEEE 802.3 Managed Object	Corresponding SNMP Object
oMacEntity	
.aMACID	dot3StatsIndex or IF-MIB - ifIndex
.aFramesTransmittedOK	IF-MIB - ifOutUCastPkts + ifOutMulticastPkts + ifOutBroadcastPkts*
.aSingleCollisionFrames	dot3StatsSingleCollisionFrames
.aMultipleCollisionFrames	dot3StatsMultipleCollisionFrames
.aFramesReceivedOK	IF-MIB - ifInUcastPkts + ifInMulticastPkts + ifInBroadcastPkts*
.aFrameCheckSequenceErrors	dot3StatsFCSErrors
.aAlignmentErrors	dot3StatsAlignmentErrors
.aOctetsTransmittedOK	IF-MIB - ifOutOctets*
.aFramesWithDeferredXmissions	dot3StatsDeferredTransmissions
.aLateCollisions	dot3StatsLateCollisions
.aFramesAbortedDueToXSColls	dot3StatsExcessiveCollisions
.aFramesLostDueToIntMACXmitError	dot3StatsInternalMacTransmitErrors
.aCarrierSenseErrors	dot3StatsCarrierSenseErrors
.aOctetsReceivedOK	IF-MIB - ifInOctets*
.aFramesLostDueToIntMACRcvError	dot3StatsInternalMacReceiveErrors
.aPromiscuousStatus	IF-MIB - ifPromiscuousMode
.aReadMulticastAddressList	IF-MIB - ifRcvAddressTable
.aMulticastFramesXmittedOK	IF-MIB - ifOutMulticastPkts*

.aBroadcastFramesXmittedOK	IF-MIB - ifOutBroadcastPkts*
.aMulticastFramesReceivedOK	IF-MIB - ifInMulticastPkts*
.aBroadcastFramesReceivedOK	IF-MIB - ifInBroadcastPkts*
.aFrameTooLongErrors	dot3StatsFrameTooLongs
.aReadWriteMACAddress	IF-MIB - ifPhysAddress
.aCollisionFrames	dot3CollFrequencies
.aDuplexStatus	dot3StatsDuplexStatus
.acAddGroupAddress	IF-MIB - ifRcvAddressTable
.acDeleteGroupAddress	IF-MIB - ifRcvAddressTable
.acExecuteSelfTest	dot3TestLoopBack
oPHYEntity	
.aPHYID	dot3StatsIndex or IF-MIB - ifIndex
.aSQETestErrors	dot3StatsSQETestErrors
.aSymbolErrorDuringCarrier	dot3StatsSymbolErrors
oMACControlEntity	
.aMACControlID	dot3StatsIndex or IF-MIB - ifIndex
.aMACControlFunctionsSupported	dot3ControlFunctionsSupported and dot3ControlFunctionsEnabled
.aUnsupportedOpcodesReceived	dot3ControlInUnknownOpcodes
oPAUSEEntity	
.aPAUSEMACCtrlFramesTransmitted	dot3OutPauseFrames
.aPAUSEMACCtrlFramesReceived	dot3InPauseFrames

* Note that the octet counters in IF-MIB do not exactly match the definition of the octet counters in IEEE 802.3. aOctetsTransmittedOK and aOctetsReceivedOK count only the octets in the clientData and Pad fields, whereas ifInOctets and ifOutOctets include the entire MAC frame, including MAC header and FCS. However, the IF-MIB counters can be derived from the IEEE 802.3 counters as follows:

$$\text{ifInOctets} = \text{aOctetsReceivedOK} + (18 * \text{aFramesReceivedOK})$$

$$\text{ifOutOctets} = \text{aOctetsTransmittedOK} + (18 * \text{aFramesTransmittedOK})$$

Also note that the packet counters in the IF-MIB do not exactly match the definition of the frame counters in IEEE 802.3. aFramesTransmittedOK counts the number of frames successfully transmitted on the interface, whereas ifOutUcastPkts, ifOutMulticastPkts and ifOutBroadcastPkts count the number of transmit requests made from a higher layer, whether or not the transmit attempt was successful. This means that packets counted by ifOutErrors or ifOutDiscards are also be counted by ifOut*castPkts, but are not be counted by aFramesTransmittedOK. This also means

that, since MAC Control frames are generated by a sublayer internal to the interface layer rather than by a higher layer, they are not counted by `ifOut*castPkts`, but are counted by `aFramesTransmittedOK`.

Similarly, `aFramesReceivedOK` counts the number of frames received successfully by the interface, whether or not they are passed to a higher layer, whereas `ifInUcastPkts`, `ifInMulticastPkts` and `ifInBroadcastPkts` count only the number of packets passed to a higher layer. This means that packets counted by `ifInDiscards` or `ifInUnknownProtos` are also counted by `aFramesReceivedOK`, but are not counted by `ifIn*castPkts`. This also means that, since MAC Control frames are consumed by a sublayer internal to the interface layer and not passed to a higher layer, they are not counted by `ifIn*castPkts`, but are counted by `aFramesReceivedOK`.

Another difference to keep in mind between the IF-MIB counters and IEEE 802.3 counters is that in the IEEE 802.3 document, the frame counters and octet counters are always incremented together. `aOctetsTransmittedOK` counts the number of octets in frames that were counted by `aFramesTransmittedOK`. `aOctetsReceivedOK` counts the number of octets in frames that were counted by `aFramesReceivedOK`. This is not the case with the IF-MIB counters. The IF-MIB octet counters count the number of octets sent to or received from the layer below this interface, whereas the packet counters count the number of packets sent to or received from the layer above. Therefore, received MAC Control frames, `ifInDiscards`, and `ifInUnknownProtos` are counted by `ifInOctets`, but not `ifIn*castPkts`. Transmitted MAC Control frames are counted by `ifOutOctets`, but not `ifOut*castPkts`. `ifOutDiscards` and `ifOutErrors` are counted by `ifOut*castPkts`, but not `ifOutOctets`.

The following IEEE 802.3 managed objects have been removed from this MIB module as a result of implementation feedback:

```
oMacEntity
  .aFramesWithExcessiveDeferral
  .aInRangeLengthErrors
  .aOutOfRangeLengthField
  .aMACEnableStatus
  .aTransmitEnableStatus
  .aMulticastReceiveStatus
  .acInitializeMAC
```

Please see [19] for the detailed reasoning on why these objects were removed.

In addition, the following IEEE 802.3 managed objects have not been included in this MIB for the following reasons.

IEEE 802.3 Managed Object	Disposition
oMACEntity .aMACCapabilities	Can be derived from MAU-MIB - ifMauTypeListBits
oPHYEntity .aPhyType	Can be derived from MAU-MIB - ifMauType
.aPhyTypeList	Can be derived from MAU-MIB - ifMauTypeListBits
.aMIIDetect	Not considered useful.
.aPhyAdminState	Can already obtain interface state from IF-MIB - ifOperStatus and MAU state from MAU-MIB - ifMauStatus. Providing an additional state for the PHY was not considered useful.
.acPhyAdminControl	Can already control interface state from IF-MIB - ifAdminStatus and MAU state from MAU-MIB - ifMauStatus. Providing separate admin control of the PHY was not considered useful.
oMACControlEntity .aMACControlFramesTransmitted	Can be determined by summing the OutFrames counters for the individual control functions
.aMACControlFramesReceived	Can be determined by summing the InFrames counters for the individual control functions
oPAUSEEntity .aPAUSELinkDelayAllowance	Not considered useful.

4. Definitions

```
EtherLike-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
  MODULE-IDENTITY, OBJECT-TYPE, OBJECT-IDENTITY,
  Counter32, mib-2, transmission
  FROM SNMPv2-SMI
  MODULE-COMPLIANCE, OBJECT-GROUP
  FROM SNMPv2-CONF
  ifIndex, InterfaceIndex
  FROM IF-MIB;
```

```
etherMIB MODULE-IDENTITY
```

```
  LAST-UPDATED "9908240400Z" -- August 24, 1999
  ORGANIZATION "IETF Ethernet Interfaces and Hub MIB
  Working Group"
```

```
CONTACT-INFO
```

```
  "WG E-mail: hubmib@hprnd.rose.hp.com
  To subscribe: hubmib-request@hprnd.rose.hp.com
```

```
  Chair: Dan Romascanu
  Postal: Lucent Technologies
  Atidum Technology Park, Bldg. 3
  Tel Aviv 61131
  Israel
  Tel: +972 3 645 8414
  E-mail: dromasca@lucent.com
```

```
  Editor: John Flick
  Postal: Hewlett-Packard Company
  8000 Foothills Blvd. M/S 5557
  Roseville, CA 95747-5557
  USA
  Tel: +1 916 785 4018
  Fax: +1 916 785 1199
  E-mail: johnf@rose.hp.com
```

```
  Editor: Jeffrey Johnson
  Postal: RedBack Networks
  2570 North First Street, Suite 410
  San Jose, CA, 95131
  USA
  Tel: +1 408 571 2699
  Fax: +1 408 571 2698
  E-Mail: jeff@redbacknetworks.com"
```

```
DESCRIPTION "The MIB module to describe generic objects for
```

Ethernet-like network interfaces.

The following reference is used throughout this MIB module:

[IEEE 802.3 Std] refers to:
 IEEE Std 802.3, 1998 Edition: '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', September 1998.

Of particular interest is Clause 30, '10Mb/s, 100Mb/s and 1000Mb/s Management'."

REVISION "9908240400Z" -- August 24, 1999
 DESCRIPTION "Updated to include support for 1000 Mb/sec interfaces and full-duplex interfaces. This version published as RFC 2665."

REVISION "9806032150Z"
 DESCRIPTION "Updated to include support for 100 Mb/sec interfaces. This version published as RFC 2358."

REVISION "9402030400Z"
 DESCRIPTION "Initial version, published as RFC 1650."

::= { mib-2 35 }

etherMIBObjects OBJECT IDENTIFIER ::= { etherMIB 1 }

dot3 OBJECT IDENTIFIER ::= { transmission 7 }

-- the Ethernet-like Statistics group

dot3StatsTable OBJECT-TYPE
 SYNTAX SEQUENCE OF Dot3StatsEntry
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION "Statistics for a collection of ethernet-like interfaces attached to a particular system. There will be one row in this table for each

```

        ethernet-like interface in the system."
 ::= { dot3 2 }

dot3StatsEntry OBJECT-TYPE
SYNTAX      Dot3StatsEntry
MAX-ACCESS not-accessible
STATUS      current
DESCRIPTION "Statistics for a particular interface to an
            ethernet-like medium."
INDEX       { dot3StatsIndex }
 ::= { dot3StatsTable 1 }

Dot3StatsEntry ::=
SEQUENCE {
    dot3StatsIndex                InterfaceIndex,
    dot3StatsAlignmentErrors      Counter32,
    dot3StatsFCSErrors            Counter32,
    dot3StatsSingleCollisionFrames Counter32,
    dot3StatsMultipleCollisionFrames Counter32,
    dot3StatsSQETestErrors        Counter32,
    dot3StatsDeferredTransmissions Counter32,
    dot3StatsLateCollisions       Counter32,
    dot3StatsExcessiveCollisions  Counter32,
    dot3StatsInternalMacTransmitErrors Counter32,
    dot3StatsCarrierSenseErrors   Counter32,
    dot3StatsFrameTooLongs        Counter32,
    dot3StatsInternalMacReceiveErrors Counter32,
    dot3StatsEtherChipSet         OBJECT IDENTIFIER,
    dot3StatsSymbolErrors         Counter32,
    dot3StatsDuplexStatus          INTEGER
}

dot3StatsIndex OBJECT-TYPE
SYNTAX      InterfaceIndex
MAX-ACCESS read-only
STATUS      current
DESCRIPTION "An index value that uniquely identifies an
            interface to an ethernet-like medium.  The
            interface identified by a particular value of
            this index is the same interface as identified
            by the same value of ifIndex."
REFERENCE   "RFC 2233, ifIndex"
 ::= { dot3StatsEntry 1 }

dot3StatsAlignmentErrors OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS read-only
STATUS      current

```

DESCRIPTION "A count of frames received on a particular interface that are not an integral number of octets in length and do not pass the FCS check.

The count represented by an instance of this object is incremented when the alignmentError status is returned by the MAC service to the LLC (or other MAC user). Received frames for which multiple error conditions obtain are, according to the conventions of IEEE 802.3 Layer Management, counted exclusively according to the error status presented to the LLC.

This counter does not increment for 8-bit wide group encoding schemes.

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

REFERENCE "[IEEE 802.3 Std.], 30.3.1.1.7, aAlignmentErrors"

::= { dot3StatsEntry 2 }

dot3StatsFCSErrors OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION "A count of frames received on a particular interface that are an integral number of octets in length but do not pass the FCS check. This count does not include frames received with frame-too-long or frame-too-short error.

The count represented by an instance of this object is incremented when the frameCheckError status is returned by the MAC service to the LLC (or other MAC user). Received frames for which multiple error conditions obtain are, according to the conventions of IEEE 802.3 Layer Management, counted exclusively according to the error status presented to the LLC.

Note: Coding errors detected by the physical layer for speeds above 10 Mb/s will cause the frame to fail the FCS check.

Discontinuities in the value of this counter can occur at re-initialization of the management

system, and at other times as indicated by the value of ifCounterDiscontinuityTime."

REFERENCE "[IEEE 802.3 Std.], 30.3.1.1.6, aFrameCheckSequenceErrors."

::= { dot3StatsEntry 3 }

dot3StatsSingleCollisionFrames OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION "A count of successfully transmitted frames on a particular interface for which transmission is inhibited by exactly one collision.

A frame that is counted by an instance of this object is also counted by the corresponding instance of either the ifOutUcastPkts, ifOutMulticastPkts, or ifOutBroadcastPkts, and is not counted by the corresponding instance of the dot3StatsMultipleCollisionFrames object.

This counter does not increment when the interface is operating in full-duplex mode.

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

REFERENCE "[IEEE 802.3 Std.], 30.3.1.1.3, aSingleCollisionFrames."

::= { dot3StatsEntry 4 }

dot3StatsMultipleCollisionFrames OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION "A count of successfully transmitted frames on a particular interface for which transmission is inhibited by more than one collision.

A frame that is counted by an instance of this object is also counted by the corresponding instance of either the ifOutUcastPkts, ifOutMulticastPkts, or ifOutBroadcastPkts, and is not counted by the corresponding instance of the dot3StatsSingleCollisionFrames object.

This counter does not increment when the interface is operating in full-duplex mode.

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

REFERENCE "[IEEE 802.3 Std.], 30.3.1.1.4,
aMultipleCollisionFrames."
 ::= { dot3StatsEntry 5 }

dot3StatsSQETestErrors OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION "A count of times that the SQE TEST ERROR message is generated by the PLS sublayer for a particular interface. The SQE TEST ERROR is set in accordance with the rules for verification of the SQE detection mechanism in the PLS Carrier Sense Function as described in IEEE Std. 802.3, 1998 Edition, section 7.2.4.6.

This counter does not increment on interfaces operating at speeds greater than 10 Mb/s, or on interfaces operating in full-duplex mode.

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

REFERENCE "[IEEE 802.3 Std.], 7.2.4.6, also 30.3.2.1.4,
aSQETestErrors."
 ::= { dot3StatsEntry 6 }

dot3StatsDeferredTransmissions OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION "A count of frames for which the first transmission attempt on a particular interface is delayed because the medium is busy. The count represented by an instance of this object does not include frames involved in collisions.

This counter does not increment when the interface is operating in full-duplex mode.

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

REFERENCE "[IEEE 802.3 Std.], 30.3.1.1.9,
aFramesWithDeferredXmissions."
 ::= { dot3StatsEntry 7 }

dot3StatsLateCollisions OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current

DESCRIPTION "The number of times that a collision is detected on a particular interface later than one slotTime into the transmission of a packet.

A (late) collision included in a count represented by an instance of this object is also considered as a (generic) collision for purposes of other collision-related statistics.

This counter does not increment when the interface is operating in full-duplex mode.

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

REFERENCE "[IEEE 802.3 Std.], 30.3.1.1.10,
aLateCollisions."
 ::= { dot3StatsEntry 8 }

dot3StatsExcessiveCollisions OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current

DESCRIPTION "A count of frames for which transmission on a particular interface fails due to excessive collisions.

This counter does not increment when the interface is operating in full-duplex mode.

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

REFERENCE "[IEEE 802.3 Std.], 30.3.1.1.11,

```

        aFramesAbortedDueToXSColls."
 ::= { dot3StatsEntry 9 }

```

dot3StatsInternalMacTransmitErrors OBJECT-TYPE

```

SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION "A count of frames for which transmission on a
particular interface fails due to an internal
MAC sublayer transmit error. A frame is only
counted by an instance of this object if it is
not counted by the corresponding instance of
either the dot3StatsLateCollisions object, the
dot3StatsExcessiveCollisions object, or the
dot3StatsCarrierSenseErrors object."

```

The precise meaning of the count represented by an instance of this object is implementation-specific. In particular, an instance of this object may represent a count of transmission errors on a particular interface that are not otherwise counted.

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

```

REFERENCE   "[IEEE 802.3 Std.], 30.3.1.1.12,
aFramesLostDueToIntMACXmitError."
 ::= { dot3StatsEntry 10 }

```

dot3StatsCarrierSenseErrors OBJECT-TYPE

```

SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION "The number of times that the carrier sense
condition was lost or never asserted when
attempting to transmit a frame on a particular
interface."

```

The count represented by an instance of this object is incremented at most once per transmission attempt, even if the carrier sense condition fluctuates during a transmission attempt.

This counter does not increment when the interface is operating in full-duplex mode.

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

REFERENCE "[IEEE 802.3 Std.], 30.3.1.1.13,
aCarrierSenseErrors."

::= { dot3StatsEntry 11 }

-- { dot3StatsEntry 12 } is not assigned

dot3StatsFrameTooLongs OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION "A count of frames received on a particular interface that exceed the maximum permitted frame size.

The count represented by an instance of this object is incremented when the frameTooLong status is returned by the MAC service to the LLC (or other MAC user). Received frames for which multiple error conditions obtain are, according to the conventions of IEEE 802.3 Layer Management, counted exclusively according to the error status presented to the LLC.

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

REFERENCE "[IEEE 802.3 Std.], 30.3.1.1.25,
aFrameTooLongErrors."

::= { dot3StatsEntry 13 }

-- { dot3StatsEntry 14 } is not assigned

-- { dot3StatsEntry 15 } is not assigned

dot3StatsInternalMacReceiveErrors OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION "A count of frames for which reception on a particular interface fails due to an internal MAC sublayer receive error. A frame is only counted by an instance of this object if it is not counted by the corresponding instance of

either the dot3StatsFrameTooLongs object, the dot3StatsAlignmentErrors object, or the dot3StatsFCSErrors object.

The precise meaning of the count represented by an instance of this object is implementation-specific. In particular, an instance of this object may represent a count of receive errors on a particular interface that are not otherwise counted.

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

REFERENCE "[IEEE 802.3 Std.], 30.3.1.1.15,
aFramesLostDueToIntMACRcvError."
 ::= { dot3StatsEntry 16 }

dot3StatsEtherChipSet OBJECT-TYPE
SYNTAX OBJECT IDENTIFIER
MAX-ACCESS read-only
STATUS deprecated
DESCRIPTION "***** THIS OBJECT IS DEPRECATED *****"

This object contains an OBJECT IDENTIFIER which identifies the chipset used to realize the interface. Ethernet-like interfaces are typically built out of several different chips. The MIB implementor is presented with a decision of which chip to identify via this object. The implementor should identify the chip which is usually called the Medium Access Control chip. If no such chip is easily identifiable, the implementor should identify the chip which actually gathers the transmit and receive statistics and error indications. This would allow a manager station to correlate the statistics and the chip generating them, giving it the ability to take into account any known anomalies in the chip."

::= { dot3StatsEntry 17 }

dot3StatsSymbolErrors OBJECT-TYPE
SYNTAX Counter32

MAX-ACCESS read-only
 STATUS current
 DESCRIPTION "For an interface operating at 100 Mb/s, the number of times there was an invalid data symbol when a valid carrier was present.

For an interface operating in half-duplex mode at 1000 Mb/s, the number of times the receiving media is non-idle (a carrier event) for a period of time equal to or greater than slotTime, and during which there was at least one occurrence of an event that causes the PHY to indicate 'Data reception error' or 'carrier extend error' on the GMII.

For an interface operating in full-duplex mode at 1000 Mb/s, the number of times the receiving media is non-idle (a carrier event) for a period of time equal to or greater than minFrameSize, and during which there was at least one occurrence of an event that causes the PHY to indicate 'Data reception error' on the GMII.

The count represented by an instance of this object is incremented at most once per carrier event, even if multiple symbol errors occur during the carrier event. This count does not increment if a collision is present.

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

REFERENCE "[IEEE 802.3 Std.], 30.3.2.1.5, aSymbolErrorDuringCarrier."
 ::= { dot3StatsEntry 18 }

dot3StatsDuplexStatus OBJECT-TYPE

SYNTAX INTEGER {
 unknown(1),
 halfDuplex(2),
 fullDuplex(3)
 }

MAX-ACCESS read-only

STATUS current

DESCRIPTION "The current mode of operation of the MAC entity. 'unknown' indicates that the current duplex mode could not be determined.

Management control of the duplex mode is accomplished through the MAU MIB. When an interface does not support autonegotiation, or when autonegotiation is not enabled, the duplex mode is controlled using ifMauDefaultType. When autonegotiation is supported and enabled, duplex mode is controlled using ifMauAutoNegAdvertisedBits. In either case, the currently operating duplex mode is reflected both in this object and in ifMauType.

Note that this object provides redundant information with ifMauType. Normally, redundant objects are discouraged. However, in this instance, it allows a management application to determine the duplex status of an interface without having to know every possible value of ifMauType. This was felt to be sufficiently valuable to justify the redundancy."

REFERENCE "[IEEE 802.3 Std.], 30.3.1.1.32,
aDuplexStatus."

::= { dot3StatsEntry 19 }

-- the Ethernet-like Collision Statistics group

-- Implementation of this group is optional; it is appropriate
-- for all systems which have the necessary metering

dot3CollTable OBJECT-TYPE

SYNTAX SEQUENCE OF Dot3CollEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION "A collection of collision histograms for a particular set of interfaces."

REFERENCE "[IEEE 802.3 Std.], 30.3.1.1.30,
aCollisionFrames."

::= { dot3 5 }

dot3CollEntry OBJECT-TYPE

SYNTAX Dot3CollEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION "A cell in the histogram of per-frame collisions for a particular interface. An instance of this object represents the frequency of individual MAC frames for which the transmission (successful or otherwise) on a particular interface is accompanied by a

```

        particular number of media collisions."
INDEX      { ifIndex, dot3CollCount }
 ::= { dot3CollTable 1 }

Dot3CollEntry ::=
SEQUENCE {
    dot3CollCount          INTEGER,
    dot3CollFrequencies    Counter32
}

-- { dot3CollEntry 1 } is no longer in use

dot3CollCount OBJECT-TYPE
SYNTAX      INTEGER (1..16)
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "The number of per-frame media collisions for
            which a particular collision histogram cell
            represents the frequency on a particular
            interface."
 ::= { dot3CollEntry 2 }

dot3CollFrequencies OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION "A count of individual MAC frames for which the
            transmission (successful or otherwise) on a
            particular interface occurs after the
            frame has experienced exactly the number
            of collisions in the associated
            dot3CollCount object.

            For example, a frame which is transmitted
            on interface 77 after experiencing
            exactly 4 collisions would be indicated
            by incrementing only dot3CollFrequencies.77.4.
            No other instance of dot3CollFrequencies would
            be incremented in this example.

            This counter does not increment when the
            interface is operating in full-duplex mode.

            Discontinuities in the value of this counter can
            occur at re-initialization of the management
            system, and at other times as indicated by the
            value of ifCounterDiscontinuityTime."
 ::= { dot3CollEntry 3 }

```

```

dot3ControlTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Dot3ControlEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "A table of descriptive and status information
                about the MAC Control sublayer on the
                ethernet-like interfaces attached to a
                particular system.  There will be one row in
                this table for each ethernet-like interface in
                the system which implements the MAC Control
                sublayer.  If some, but not all, of the
                ethernet-like interfaces in the system implement
                the MAC Control sublayer, there will be fewer
                rows in this table than in the dot3StatsTable."
    ::= { dot3 9 }

dot3ControlEntry OBJECT-TYPE
    SYNTAX      Dot3ControlEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "An entry in the table, containing information
                about the MAC Control sublayer on a single
                ethernet-like interface."
    INDEX       { dot3StatsIndex }
    ::= { dot3ControlTable 1 }

Dot3ControlEntry ::=
    SEQUENCE {
        dot3ControlFunctionsSupported      BITS,
        dot3ControlInUnknownOpcodes      Counter32
    }

dot3ControlFunctionsSupported OBJECT-TYPE
    SYNTAX      BITS {
        pause(0)      -- 802.3x flow control
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "A list of the possible MAC Control functions
                implemented for this interface."
    REFERENCE   "[IEEE 802.3 Std.], 30.3.3.2,
                aMACControlFunctionsSupported."
    ::= { dot3ControlEntry 1 }

dot3ControlInUnknownOpcodes OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current

```

DESCRIPTION "A count of MAC Control frames received on this interface that contain an opcode that is not supported by this device.

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

REFERENCE "[IEEE 802.3 Std.], 30.3.3.5,
aUnsupportedOpCodesReceived"

::= { dot3ControlEntry 2 }

dot3PauseTable OBJECT-TYPE

SYNTAX SEQUENCE OF Dot3PauseEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION "A table of descriptive and status information about the MAC Control PAUSE function on the ethernet-like interfaces attached to a particular system. There will be one row in this table for each ethernet-like interface in the system which supports the MAC Control PAUSE function (i.e., the 'pause' bit in the corresponding instance of dot3ControlFunctionsSupported is set). If some, but not all, of the ethernet-like interfaces in the system implement the MAC Control PAUSE function (for example, if some interfaces only support half-duplex), there will be fewer rows in this table than in the dot3StatsTable."

::= { dot3 10 }

dot3PauseEntry OBJECT-TYPE

SYNTAX Dot3PauseEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION "An entry in the table, containing information about the MAC Control PAUSE function on a single ethernet-like interface."

INDEX { dot3StatsIndex }

::= { dot3PauseTable 1 }

Dot3PauseEntry ::=

SEQUENCE {

dot3PauseAdminMode	INTEGER,
dot3PauseOperMode	INTEGER,
dot3InPauseFrames	Counter32,
dot3OutPauseFrames	Counter32

}

```

dot3PauseAdminMode OBJECT-TYPE
  SYNTAX      INTEGER {
                disabled(1),
                enabledXmit(2),
                enabledRcv(3),
                enabledXmitAndRcv(4)
              }
  MAX-ACCESS  read-write
  STATUS      current
  DESCRIPTION "This object is used to configure the default
              administrative PAUSE mode for this interface.

```

This object represents the administratively-configured PAUSE mode for this interface. If auto-negotiation is not enabled or is not implemented for the active MAU attached to this interface, the value of this object determines the operational PAUSE mode of the interface whenever it is operating in full-duplex mode. In this case, a set to this object will force the interface into the specified mode.

If auto-negotiation is implemented and enabled for the MAU attached to this interface, the PAUSE mode for this interface is determined by auto-negotiation, and the value of this object denotes the mode to which the interface will automatically revert if/when auto-negotiation is later disabled. Note that when auto-negotiation is running, administrative control of the PAUSE mode may be accomplished using the ifMauAutoNegCapAdvertisedBits object in the MAU-MIB.

Note that the value of this object is ignored when the interface is not operating in full-duplex mode.

An attempt to set this object to 'enabledXmit(2)' or 'enabledRcv(3)' will fail on interfaces that do not support operation at greater than 100 Mb/s."

```
 ::= { dot3PauseEntry 1 }
```

```

dot3PauseOperMode OBJECT-TYPE

```

```

SYNTAX      INTEGER {
                disabled(1),
                enabledXmit(2),
                enabledRcv(3),
                enabledXmitAndRcv(4)
            }
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION "This object reflects the PAUSE mode currently
            in use on this interface, as determined by
            either (1) the result of the auto-negotiation
            function or (2) if auto-negotiation is not
            enabled or is not implemented for the active MAU
            attached to this interface, by the value of
            dot3PauseAdminMode. Interfaces operating at
            100 Mb/s or less will never return
            'enabledXmit(2)' or 'enabledRcv(3)'. Interfaces
            operating in half-duplex mode will always return
            'disabled(1)'. Interfaces on which
            auto-negotiation is enabled but not yet
            completed should return the value
            'disabled(1)'."
 ::= { dot3PauseEntry 2 }

```

dot3InPauseFrames OBJECT-TYPE

```

SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION "A count of MAC Control frames received on this
            interface with an opcode indicating the PAUSE
            operation.

```

This counter does not increment when the interface is operating in half-duplex mode. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."

```

REFERENCE   "[IEEE 802.3 Std.], 30.3.4.3,
            aPAUSEMACCtrlFramesReceived."
 ::= { dot3PauseEntry 3 }

```

dot3OutPauseFrames OBJECT-TYPE

```

SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION "A count of MAC Control frames transmitted on
            this interface with an opcode indicating the

```

PAUSE operation.

This counter does not increment when the interface is operating in half-duplex mode.

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

REFERENCE "[IEEE 802.3 Std.], 30.3.4.2,
aPAUSEMACCtrlFramesTransmitted."

::= { dot3PauseEntry 4 }

-- 802.3 Tests

dot3Tests OBJECT IDENTIFIER ::= { dot3 6 }

dot3Errors OBJECT IDENTIFIER ::= { dot3 7 }

-- TDR Test

dot3TestTdr OBJECT-IDENTITY

STATUS current

DESCRIPTION "The Time-Domain Reflectometry (TDR) test is specific to ethernet-like interfaces of type 10Base5 and 10Base2. The TDR value may be useful in determining the approximate distance to a cable fault. It is advisable to repeat this test to check for a consistent resulting TDR value, to verify that there is a fault.

A TDR test returns as its result the time interval, measured in 10 MHz ticks or 100 nsec units, between the start of TDR test transmission and the subsequent detection of a collision or deassertion of carrier. On successful completion of a TDR test, the result is stored as the value of an appropriate instance of an appropriate vendor specific MIB object, and the OBJECT IDENTIFIER of that instance is stored in the appropriate instance of the appropriate test result code object (thereby indicating where the result has been stored)."

::= { dot3Tests 1 }

-- Loopback Test

```

dot3TestLoopBack OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION "This test configures the MAC chip and executes
              an internal loopback test of memory, data paths,
              and the MAC chip logic. This loopback test can
              only be executed if the interface is offline.
              Once the test has completed, the MAC chip should
              be reinitialized for network operation, but it
              should remain offline.

              If an error occurs during a test, the
              appropriate test result object will be set
              to indicate a failure. The two OBJECT
              IDENTIFIER values dot3ErrorInitError and
              dot3ErrorLoopbackError may be used to provided
              more information as values for an appropriate
              test result code object."
  ::= { dot3Tests 2 }

dot3ErrorInitError OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION "Couldn't initialize MAC chip for test."
  ::= { dot3Errors 1 }

dot3ErrorLoopbackError OBJECT-IDENTITY
  STATUS      current
  DESCRIPTION "Expected data not received (or not received
              correctly) in loopback test."
  ::= { dot3Errors 2 }

-- { dot3 8 }, the dot3ChipSets tree, is defined in [28]

-- conformance information

etherConformance OBJECT IDENTIFIER ::= { etherMIB 2 }

etherGroups      OBJECT IDENTIFIER ::= { etherConformance 1 }
etherCompliances OBJECT IDENTIFIER ::= { etherConformance 2 }

-- compliance statements

etherCompliance MODULE-COMPLIANCE
  STATUS      deprecated
  DESCRIPTION "***** THIS COMPLIANCE IS DEPRECATED *****

              The compliance statement for managed network
              entities which have ethernet-like network
              interfaces.

```

This compliance is deprecated and replaced by dot3Compliance."

```

MODULE -- this module
  MANDATORY-GROUPS { etherStatsGroup }

  GROUP      etherCollisionTableGroup
  DESCRIPTION "This group is optional. It is appropriate
               for all systems which have the necessary
               metering. Implementation in such systems is
               highly recommended."
 ::= { etherCompliances 1 }

ether100MbsCompliance MODULE-COMPLIANCE
  STATUS      deprecated
  DESCRIPTION "***** THIS COMPLIANCE IS DEPRECATED *****

               The compliance statement for managed network
               entities which have 100 Mb/sec ethernet-like
               network interfaces.

               This compliance is deprecated and replaced by
               dot3Compliance."

MODULE -- this module
  MANDATORY-GROUPS { etherStats100MbsGroup }

  GROUP      etherCollisionTableGroup
  DESCRIPTION "This group is optional. It is appropriate
               for all systems which have the necessary
               metering. Implementation in such systems is
               highly recommended."
 ::= { etherCompliances 2 }

dot3Compliance MODULE-COMPLIANCE
  STATUS      current
  DESCRIPTION "The compliance statement for managed network
               entities which have ethernet-like network
               interfaces."

MODULE -- this module
  MANDATORY-GROUPS { etherStatsBaseGroup }

  GROUP      etherDuplexGroup
  DESCRIPTION "This group is mandatory for all
               ethernet-like network interfaces which are
               capable of operating in full-duplex mode.
               It is highly recommended for all

```

ethernet-like network interfaces."

GROUP etherStatsLowSpeedGroup
 DESCRIPTION "This group is mandatory for all ethernet-like network interfaces which are capable of operating at 10 Mb/s or slower in half-duplex mode."

GROUP etherStatsHighSpeedGroup
 DESCRIPTION "This group is mandatory for all ethernet-like network interfaces which are capable of operating at 100 Mb/s or faster."

GROUP etherControlGroup
 DESCRIPTION "This group is mandatory for all ethernet-like network interfaces that support the MAC Control sublayer."

GROUP etherControlPauseGroup
 DESCRIPTION "This group is mandatory for all ethernet-like network interfaces that support the MAC Control PAUSE function."

GROUP etherCollisionTableGroup
 DESCRIPTION "This group is optional. It is appropriate for all ethernet-like network interfaces which are capable of operating in half-duplex mode and have the necessary metering. Implementation in systems with such interfaces is highly recommended."

::= { etherCompliances 3 }

-- units of conformance

etherStatsGroup OBJECT-GROUP
 OBJECTS { dot3StatsIndex,
 dot3StatsAlignmentErrors,
 dot3StatsFCSErrors,
 dot3StatsSingleCollisionFrames,
 dot3StatsMultipleCollisionFrames,
 dot3StatsSQETestErrors,
 dot3StatsDeferredTransmissions,
 dot3StatsLateCollisions,
 dot3StatsExcessiveCollisions,
 dot3StatsInternalMacTransmitErrors,
 dot3StatsCarrierSenseErrors,
 dot3StatsFrameTooLongs,

```

        dot3StatsInternalMacReceiveErrors,
        dot3StatsEtherChipSet
    }
STATUS      deprecated
DESCRIPTION "***** THIS GROUP IS DEPRECATED *****"

        A collection of objects providing information
        applicable to all ethernet-like network
        interfaces.

        This object group has been deprecated and
        replaced by etherStatsBaseGroup and
        etherStatsLowSpeedGroup."
 ::= { etherGroups 1 }

etherCollisionTableGroup OBJECT-GROUP
OBJECTS      { dot3CollFrequencies
               }
STATUS      current
DESCRIPTION "A collection of objects providing a histogram
of packets successfully transmitted after
experiencing exactly N collisions."
 ::= { etherGroups 2 }

etherStats100MbsGroup OBJECT-GROUP
OBJECTS      { dot3StatsIndex,
               dot3StatsAlignmentErrors,
               dot3StatsFCSErrors,
               dot3StatsSingleCollisionFrames,
               dot3StatsMultipleCollisionFrames,
               dot3StatsDeferredTransmissions,
               dot3StatsLateCollisions,
               dot3StatsExcessiveCollisions,
               dot3StatsInternalMacTransmitErrors,
               dot3StatsCarrierSenseErrors,
               dot3StatsFrameTooLongs,
               dot3StatsInternalMacReceiveErrors,
               dot3StatsEtherChipSet,
               dot3StatsSymbolErrors
               }
STATUS      deprecated
DESCRIPTION "***** THIS GROUP IS DEPRECATED *****"

        A collection of objects providing information
        applicable to 100 Mb/sec ethernet-like network
        interfaces.

        This object group has been deprecated and

```

```

                replaced by etherStatsBaseGroup and
                etherStatsHighSpeedGroup."
 ::= { etherGroups 3 }

etherStatsBaseGroup OBJECT-GROUP
  OBJECTS      { dot3StatsIndex,
                 dot3StatsAlignmentErrors,
                 dot3StatsFCSErrors,
                 dot3StatsSingleCollisionFrames,
                 dot3StatsMultipleCollisionFrames,
                 dot3StatsDeferredTransmissions,
                 dot3StatsLateCollisions,
                 dot3StatsExcessiveCollisions,
                 dot3StatsInternalMacTransmitErrors,
                 dot3StatsCarrierSenseErrors,
                 dot3StatsFrameTooLongs,
                 dot3StatsInternalMacReceiveErrors
               }
  STATUS       current
  DESCRIPTION  "A collection of objects providing information
                applicable to all ethernet-like network
                interfaces."
 ::= { etherGroups 4 }

etherStatsLowSpeedGroup OBJECT-GROUP
  OBJECTS      { dot3StatsSQETestErrors }
  STATUS       current
  DESCRIPTION  "A collection of objects providing information
                applicable to ethernet-like network interfaces
                capable of operating at 10 Mb/s or slower in
                half-duplex mode."

 ::= { etherGroups 5 }

etherStatsHighSpeedGroup OBJECT-GROUP
  OBJECTS      { dot3StatsSymbolErrors }
  STATUS       current
  DESCRIPTION  "A collection of objects providing information
                applicable to ethernet-like network interfaces
                capable of operating at 100 Mb/s or faster."
 ::= { etherGroups 6 }

etherDuplexGroup OBJECT-GROUP
  OBJECTS      { dot3StatsDuplexStatus }
  STATUS       current
  DESCRIPTION  "A collection of objects providing information
                about the duplex mode of an ethernet-like
                network interface."

```

```

 ::= { etherGroups 7 }

etherControlGroup OBJECT-GROUP
  OBJECTS      { dot3ControlFunctionsSupported,
                 dot3ControlInUnknownOpcodes
               }
  STATUS       current
  DESCRIPTION  "A collection of objects providing information
               about the MAC Control sublayer on ethernet-like
               network interfaces."
 ::= { etherGroups 8 }

etherControlPauseGroup OBJECT-GROUP
  OBJECTS      { dot3PauseAdminMode,
                 dot3PauseOperMode,
                 dot3InPauseFrames,
                 dot3OutPauseFrames
               }
  STATUS       current
  DESCRIPTION  "A collection of objects providing information
               about and control of the MAC Control PAUSE
               function on ethernet-like network interfaces."
 ::= { etherGroups 9 }

END

```

5. Intellectual Property

The IETF takes no position regarding the validity or scope of any intellectual property 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; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication 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 implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

6. Acknowledgements

This document was produced by the IETF Ethernet Interfaces and Hub MIB Working Group, whose efforts were greatly advanced by the contributions of the following people:

Lynn Kubinec
Steve McRobert
Dan Romascanu
Andrew Smith
Geoff Thompson

This document is based on the Proposed Standard Ethernet MIB, RFC 2358 [23], edited by John Flick of Hewlett-Packard and Jeffrey Johnson of RedBack Networks and produced by the 802.3 Hub MIB Working Group. It extends that document by providing support for full-duplex Ethernet interfaces and 1000 Mb/sec Ethernet interfaces as outlined in [16].

RFC 2358, in turn, is almost completely based on both the Standard Ethernet MIB, RFC 1643 [21], and the Proposed Standard Ethernet MIB using the SNMPv2 SMI, RFC 1650 [22], both of which were edited by Frank Kastenholz of FTP Software and produced by the Interfaces MIB Working Group. RFC 2358 extends those documents by providing support for 100 Mb/sec ethernet interfaces.

RFC 1643 and RFC 1650, in turn, are based on the Draft Standard Ethernet MIB, RFC 1398 [20], also edited by Frank Kastenholz and produced by the Ethernet MIB Working Group.

RFC 1398, in turn, is based on the Proposed Standard Ethernet MIB, RFC 1284 [18], which was edited by John Cook of Chipcom and produced by the Transmission MIB Working Group. The Ethernet MIB Working Group gathered implementation experience of the variables specified in RFC 1284, documented that experience in RFC 1369 [19], and used that information to develop this revised MIB.

RFC 1284, in turn, is based on a document written by Frank Kastenholz, then of Interlan, entitled IEEE 802.3 Layer Management Draft M compatible MIB for TCP/IP Networks [17]. This document was modestly reworked, initially by the SNMP Working Group, and then by the Transmission Working Group, to reflect the current conventions for defining objects for MIB interfaces. James Davin, of the MIT Laboratory for Computer Science, and Keith McCloghrie of Hughes LAN Systems, contributed to later drafts of this memo. Marshall Rose of Performance Systems International, Inc. converted the document into

RFC 1212 [3] concise format. Anil Rijsinghani of DEC contributed text that more adequately describes the TDR test. Thanks to Frank Kastenholz of Interlan and Louis Steinberg of IBM for their experimentation.

7. References

- [1] Harrington, D., Presuhn, R. and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC 2571, May 1999.
- [2] Rose, M. and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", STD 16, RFC 1155, May 1990.
- [3] Rose, M. and K. McCloghrie, "Concise MIB Definitions", STD 16, RFC 1212, March 1991.
- [4] Rose, M., "A Convention for Defining Traps for use with the SNMP", RFC 1215, March 1991.
- [5] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Structure of Management Information Version 2 (SMIV2)", STD 58, RFC 2578, April 1999.
- [6] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Textual Conventions for SMIV2", STD 58, RFC 2579, April 1999.
- [7] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Conformance Statements for SMIV2", STD 58, RFC 2580, April 1999.
- [8] Case, J., Fedor, M., Schoffstall, M. and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, May 1990.
- [9] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, January 1996.
- [10] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1906, January 1996.
- [11] Case, J., Harrington D., Presuhn R. and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2572, May 1999.

- [12] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2574, May 1999.
- [13] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, January 1996.
- [14] Levi, D., Meyer, P. and B. Stewart, "SNMPv3 Applications", RFC 2573, May 1999.
- [15] Wijnen, B., Presuhn, R. and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2575, May 1999.
- [16] IEEE, IEEE Std 802.3, 1998 Edition: "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" (incorporating ANSI/IEEE Std. 802.3, 1996 Edition, IEEE Std. 802.3r-1996, 802.3u-1995, 802.3x&y-1997, 802.3z-1998, and 802.3aa-1998), September 1998.
- [17] Kastenholz, F., "IEEE 802.3 Layer Management Draft compatible MIB for TCP/IP Networks", electronic mail message to mib-wg@nnsf.net, 9 June 1989.
- [18] Cook, J., "Definitions of Managed Objects for Ethernet-Like Interface Types", RFC 1284, December 1991.
- [19] Kastenholz, F., "Implementation Notes and Experience for The Internet Ethernet MIB", RFC 1369, October 1992.
- [20] Kastenholz, F., "Definitions of Managed Objects for the Ethernet-like Interface Types", RFC 1398, January 1993.
- [21] Kastenholz, F., "Definitions of Managed Objects for the Ethernet-like Interface Types", STD 50, RFC 1643, July 1994.
- [22] Kastenholz, F., "Definitions of Managed Objects for the Ethernet-like Interface Types using SMIV2", RFC 1650, August 1994.
- [23] Flick, J. and J. Johnson, "Definitions of Managed Objects for the Ethernet-like Interface Types", RFC 2358, June 1998.

- [24] McCloghrie, K. and M. Rose, Editors, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", STD 17, RFC 1213, March 1991.
- [25] McCloghrie, K., and F. Kastenholz, "The Interfaces Group MIB using SMIV2", RFC 2233, November 1997.
- [26] Bradner, S., "Key words for use in RFCs to Indicate Requirements Levels", BCP 14, RFC 2119, March 1997.
- [27] Smith, A., Flick, J., deGraaf, K., Romascanu, D., McMaster, D., McCloghrie, K. and S. Roberts, "Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs)", RFC 2668, August 1999.
- [28] Flick, J., "Definitions of Object Identifiers for Identifying Ethernet Chip Sets", RFC 2666, August 1999.

8. Security Considerations

There are two management objects defined in this MIB that have a MAX-ACCESS clause of read-write. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations.

There are a number of managed objects in this MIB that may be considered to contain sensitive information. In particular, the dot3StatsEtherChipSet object may be considered sensitive in many environments, since it would allow an intruder to obtain information about which vendor's equipment is in use on the network. Note that this object has been deprecated. However, some implementors may still choose to implement it for backwards compatibility.

Therefore, it may be important in some environments to control read access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP. Not all versions of SNMP provide features for such a secure environment.

SNMPv1 by itself is such an insecure environment. 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 (read) the objects in this MIB.

It is recommended that the implementors consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model RFC 2574 [12] and the View-based Access Control Model RFC 2575 [15] is recommended.

It is then a customer/user responsibility to ensure that the SNMP entity giving access to an instance of this MIB, is properly configured to give access to those objects only to those principals (users) that have legitimate rights to access them.

9. Authors' Addresses

John Flick
Hewlett-Packard Company
8000 Foothills Blvd. M/S 5557
Roseville, CA 95747-5557

Phone: +1 916 785 4018
EMail: johnf@rose.hp.com

Jeffrey Johnson
RedBack Networks
2570 North First Street, Suite 410
San Jose, CA, 95131, USA

Phone: +1 408 571 2699
EMail: jeff@redbacknetworks.com

A. Change Log

A.1. Changes since RFC 2358

This section enumerates changes made to RFC 2358 to produce this document.

- (1) Section 2 has been replaced with the current SNMP Management Framework boilerplate.
- (2) The ifMtu mapping has been clarified.
- (3) The relationship between the IEEE 802.3 octet counters and the IF-MIB octet counters has been clarified.
- (4) REFERENCE clauses have been updated to reflect the actual IEEE 802.3 managed object that each MIB object is based on.
- (5) The following object DESCRIPTION clauses have been updated to reflect that they do not increment in full-duplex mode: dot3StatsSingleCollisionFrames, dot3StatsMultipleCollisionFrames, dot3StatsSQETestErrors, dot3StatsDeferredTransmissions, dot3StatsLateCollisions, dot3StatsExcessiveCollisions, dot3StatsCarrierSenseErrors, dot3CollFrequencies.
- (6) The following object DESCRIPTION clauses have been updated to reflect behaviour on full-duplex and 1000 Mb/s interfaces: dot3StatsAlignmentErrors, dot3StatsFCSErrors, dot3StatsSQETestErrors, dot3StatsLateCollisions, dot3StatsSymbolErrors.
- (7) Two new tables, dot3ControlTable and dot3PauseTable, have been added.
- (8) A new object, dot3StatsDuplexStatus, has been added.
- (9) The object groups and compliances have been restructured.
- (10) The dot3StatsEtherChipSet object has been deprecated.
- (11) The dot3ChipSets have been moved to a separate document.

A.2. Changes between RFC 1650 and RFC 2358

This section enumerates changes made to RFC 1650 to produce RFC 2358.

- (1) The MODULE-IDENTITY has been updated to reflect the changes in the MIB.
- (2) A new object, dot3StatsSymbolErrors, has been added.
- (3) The definition of the object dot3StatsIndex has been converted to use the SMIV2 OBJECT-TYPE macro.
- (4) A new conformance group, etherStats100MbsGroup, has been added.
- (5) A new compliance statement, ether100MbsCompliance, has been added.
- (6) The Acknowledgements were extended to provide a more complete history of the origin of this document.
- (7) The discussion of ifType has been expanded.
- (8) A section on mapping of Interfaces MIB objects has been added.
- (9) A section defining the relationship of this MIB to the MAU MIB has been added.
- (10) A section on the mapping of IEEE 802.3 managed objects to this MIB and the Interfaces MIB has been added.
- (11) Converted the dot3Tests, dot3Errors, and dot3ChipSets OIDs to use the OBJECT-IDENTITY macro.
- (12) Added to the list of registered dot3ChipSets.
- (13) An intellectual property notice and copyright notice were added, as required by RFC 2026.

B. Full Copyright Statement

Copyright (C) The Internet Society (1999). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS 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.

