Network Working Group
Request for Comments: 1284

J. Cook, Editor
Chipcom Corporation

December 1991

Definitions of Managed Objects for the Ethernet-like Interface Types

Status of this Memo

This memo is an extension to the SNMP MIB. This RFC specifies an IAB standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "IAB Official Protocol Standards" for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Table of Contents

1. Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in TCP/IP-based internets. In particular, it defines objects for managing ethernet-like objects.

2. The Network Management Framework

The Internet-standard Network Management Framework consists of three components. They are:

RFC 1155 which defines the SMI, the mechanisms used for describing and naming objects for the purpose of management. RFC 1212

defines a more concise description mechanism, which is wholly consistent with the SMI.

RFC 1156 which defines MIB-I, the core set of managed objects for the Internet suite of protocols. RFC 1213, defines MIB-II, an evolution of MIB-I based on implementation experience and new operational requirements.

RFC 1157 which defines the SNMP, the protocol used for network access to managed objects.

The Framework permits new objects to be defined for the purpose of experimentation and evaluation.

3. Objects

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the subset of Abstract Syntax Notation One (ASN.1) [7] defined in the SMI. In particular, each object has a name, a syntax, and an encoding. The name is an object identifier, an administratively assigned name, which specifies an object type. The object type together with an object instance serves to uniquely identify a specific instantiation of the object. For human convenience, we often use a textual string, termed the OBJECT DESCRIPTOR, to also refer to the object type.

The syntax of an object type defines the abstract data structure corresponding to that object type. The ASN.1 language is used for this purpose. However, the SMI [3] purposely restricts the ASN.1 constructs which may be used. These restrictions are explicitly made for simplicity.

The encoding of an object type is simply how that object type is represented using the object type's syntax. Implicitly tied to the notion of an object type's syntax and encoding is how the object type is represented when being transmitted on the network.

The SMI specifies the use of the basic encoding rules of ASN.1 [8], subject to the additional requirements imposed by the SNMP.

3.1. Format of Definitions

Section 5 contains contains the specification of all object types contained in this MIB module. The object types are defined using the conventions defined in the SMI, as amended by the extensions specified in [13].

4. 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 three values of the ifType object in the Internet-standard MIB:

```
ethernet-csmacd(6)
iso88023-csmacd(7)
starLan(11)
```

For these interfaces, the value of the ifSpecific variable in the MIB-II [6] has the OBJECT IDENTIFIER value:

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

The definitions presented here are based on the IEEE 802.3 Layer Management Specification [9], as originally interpreted by Frank Kastenholz of Interlan in [10]. Implementors of these MIB objects should note that the IEEE document 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 [9] are represented by previously defined objects in the Internet-standard MIB or in the generic interface extensions MIB [11], 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.

The relationship between an ethernet-like interface and an interface in the context of the Internet-standard MIB 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.

5. Definitions

```
    RFC1284-MIB DEFINITIONS ::= BEGIN

    IMPORTS
            Counter, Gauge
                    FROM RFC1155-SMI
            transmission
                    FROM RFC1213-MIB
            OBJECT-TYPE
                    FROM RFC-1212;

    --  This MIB module uses the extended OBJECT-TYPE macro as
    --  defined in [13]


    --  this is the MIB module for ethernet-like objects

    dot3    OBJECT IDENTIFIER ::= { transmission 7 }


    -- the Generic Ethernet-like group

    -- Implementation of this group is mandatory for all systems
    -- that attach to an ethernet-like medium.

    dot3Table OBJECT-TYPE
        SYNTAX  SEQUENCE OF Dot3Entry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "Status information and control variables for a
                collection of ethernet-like interfaces attached to
                a particular system."
        ::= { dot3 1 }

    dot3Entry OBJECT-TYPE
        SYNTAX  Dot3Entry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "Status information and control variables for a
                particular interface to an ethernet-like medium."
        INDEX   { dot3Index }
        ::= { dot3Table 1 }
```

```
    Dot3Entry ::=
        SEQUENCE {
            dot3Index
                INTEGER,
            dot3InitializeMac
                INTEGER,
            dot3MacSubLayerStatus
                INTEGER,
            dot3MulticastReceiveStatus
                INTEGER,
            dot3TxEnabled
                INTEGER,
            dot3TestTdrValue
                Gauge
       }

    dot3Index OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        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."
        ::= { dot3Entry 1 }

    dot3InitializeMac OBJECT-TYPE
        SYNTAX  INTEGER { initialized(1), uninitialized(2) }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The initialization status of the MAC and PLS
                (Physical Layer Signalling) subsystems for a
                particular interface. The value initialized(1)
                signifies that the subsystems for a particular
                interface have been previously initialized; the
                value uninitialized(2) signifies that they have
                not been previously initialized.

                Each alteration of an instance of this object to
                either of the values initialized(1) or
                uninitialized(2) is analogous to an invocation of
                the initializeMAC action defined in [9] and has
                the effect of (re-)initializing the MAC and PLS
                subsystems for the associated interface. In
                particular,
```

```
                    all management counters pertaining to the MAC
                    and PLS subsystems for said interface are
                    reset to zero;

                    the receive and transmit layer management
                    state variables (receiveEnabled and
                    transmitEnabled in [9]) are set to enable
                    reception and transmission of frames;

                    the promiscuous receive function is disabled;
                    and

                    multicast reception is disabled."
        ::= { dot3Entry 2 }

    dot3MacSubLayerStatus OBJECT-TYPE
        SYNTAX  INTEGER { enabled(1), disabled(2) }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The operational status of the MAC sublayer for a
                particular interface.  The value enabled(1)
                signifies that the MAC sublayer for said interface
                is operational for both transmitting and receiving
                frames -- that is, that the value of both the
                receive and transmit layer management state
                variables (receiveEnabled and transmitEnabled in
                [9]) for said interface are true.  The value
                disabled(2) signifies that the MAC sublayer for
                said interface is not operational for either
                transmitting or receiving frames. In particular,
                the value of an instance of this object is
                disabled(2) whenever the value of the
                corresponding instance of the dot3Enabled object
                is false(2).

                Each alteration of an instance of this object to
                the value enabled(1) is analogous to an invocation
                of the enableMACSublayer action defined in [9] and
                has the effect of starting normal transmit and
                receive operations (from the ‘‘idle’’ state) on
                the associated interface. In particular, such an
                alteration has the effect of resetting the PLS for
                said interface and of setting the receive and
                transmit layer management state variables
                (receiveEnabled and transmitEnabled in [9]) to be
                true.
```

```
                Each alteration of an instance of this object to
                the value disabled(2) is analogous to an
                invocation of the disableMACSublayer action
                defined in [9] and has the effect of terminating
                transmit and receive operations on the associated
                interface. In particular, such an alteration has
                the effect of setting the receive and transmit
                layer management state variables (receiveEnabled
                and transmitEnabled in [9]) to be false. Any
                transmissions/receptions in progress are completed
                before operation is terminated."
        ::= { dot3Entry 3 }

    dot3MulticastReceiveStatus OBJECT-TYPE
        SYNTAX  INTEGER { enabled(1), disabled(2) }
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
                "The multicast receive status for a particular
                interface.  The value enabled(1) signifies that
                reception of multicast frames by the MAC sublayer
                is enabled on said interface. The value
                disabled(2) signifies that reception of multicast
                frames by the MAC sublayer is not enabled on said
                interface.

                Each alteration of an instance of this object to
                the value enabled(1) is analogous to an invocation
                of the enableMulticastReceive action defined in
                [9] and has the effect of enabling multicast frame
                reception on the associated interface. Actual
                reception of multicast frames is only possible on
                an interface when the values for the associated
                instances of the dot3MulticastReceiveStatus and
                dot3MacSubLayerStatus objects are enabled(1) and
                enabled(1), respectively.

                Each alteration of an instance of this object to
                the value disabled(2) is analogous to an
                invocation of the disableMulticastReceive action
                defined in [9] and has the effect of inhibiting
                multicast frame reception on the associated
                interface."
        ::= { dot3Entry 4 }

    dot3TxEnabled OBJECT-TYPE
        SYNTAX  INTEGER { true(1), false(2)  }
        ACCESS  read-write
```

```
        STATUS  mandatory
        DESCRIPTION
                "The transmit layer management state variable
                (transmitEnabled as defined in [9]) for a
                particular interface.  The value true(1) signifies
                that the MAC frame transmission is enabled on said
                interface. The value false(2) signifies that the
                MAC frame transmission is inhibited on said
                interface. In particular, the value of an instance
                of this object is false(2) whenever the value of
                the corresponding instance of the
                dot3MacSubLayerStatus object is disabled(2).

                Each alteration of an instance of this object to
                the value true(1) is analogous to an invocation of
                the enableTransmit action defined in [9] and has
                the effect of enabling MAC sublayer frame
                transmission on the associated interface. In
                particular, such an alteration has the effect of
                setting the transmit layer management state
                variable (transmitEnabled in [9]) for said
                interface to be true.

                Each alteration of an instance of this object to
                the value false(2) is analogous to an invocation
                of the disableTransmit action defined in [9] and
                has the effect of inhibiting MAC sublayer frame
                transmission on the associated interface. In
                particular, such an alteration has the effect of
                setting the transmit layer management state
                variable (transmitEnabled in [9]) for said
                interface to be false. Any transmissions in
                progress are completed before transmission is
                inhibited."
        ::= { dot3Entry 5 }

    dot3TestTdrValue OBJECT-TYPE
        SYNTAX  Gauge
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
               "The number of 10 MHz ticks which elapsed between
               the beginning of a TDR measurement and the
               collision which ended it, for the most recently
               executed TDR test.  If no TDR test has been
               executed, or the last TDR value is not available,
              this object has the value 0."
        ::= { dot3Entry 6 }
```

```
    -- the Ethernet-like Statistics group

    -- Implementation of this group is mandatory

    -- Due to implementation restrictions (e.g. in the
    -- instrumentation provided by a chipset, or a device
    -- driver), some of the counters in this group may be
    -- difficult or impossible to implement.
    -- In such cases, an implementator should apply reasonable
    -- best effort to detect as many occurrences as possible.
    -- In any case, the value of a counter will be the number
    -- actually detected, which will always be less or equal
    -- to the number of actual occurrences.  In the extreme
    -- case of a total inability to detect occurrences, the
    -- counter will always be zero.

    -- Vendors are strongly encouraged to document in user guides and
    -- other appropriate documentation the conditions under which the
    -- values of the counters in this group may represent an
    -- underestimate of the true count.

    dot3StatsTable OBJECT-TYPE
        SYNTAX  SEQUENCE OF Dot3StatsEntry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "Statistics for a collection of ethernet-like
                interfaces attached to a particular system."
        ::= { dot3 2 }

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

    Dot3StatsEntry ::=
        SEQUENCE {
            dot3StatsIndex
                INTEGER,
            dot3StatsAlignmentErrors
                Counter,
            dot3StatsFCSErrors
                Counter,
```

```
            dot3StatsSingleCollisionFrames
                Counter,
            dot3StatsMultipleCollisionFrames
                Counter,
            dot3StatsSQETestErrors
                Counter,
            dot3StatsDeferredTransmissions
                Counter,
            dot3StatsLateCollisions
                Counter,
            dot3StatsExcessiveCollisions
                Counter,
            dot3StatsInternalMacTransmitErrors
                Counter,
            dot3StatsCarrierSenseErrors
                Counter,
            dot3StatsExcessiveDeferrals
                Counter,
            dot3StatsFrameTooLongs
                Counter,
            dot3StatsInRangeLengthErrors
                Counter,
            dot3StatsOutOfRangeLengthFields
                Counter,
            dot3StatsInternalMacReceiveErrors
                Counter
        }

    dot3StatsIndex OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        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."
        ::= { dot3StatsEntry 1 }

    dot3StatsAlignmentErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        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 [9], counted exclusively
                according to the error status presented to the
                LLC."
        ::= { dot3StatsEntry 2 }

    dot3StatsFCSErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        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.

                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 [9], counted exclusively
                according to the error status presented to the
                LLC."
        ::= { dot3StatsEntry 3 }

    dot3StatsSingleCollisionFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        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 or
                ifOutNUcastPkts object and is not counted by the
                corresponding instance of the
                dot3StatsMultipleCollisionFrames object."
       ::= { dot3StatsEntry 4 }
```

```
    dot3StatsMultipleCollisionFrames OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        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 or
                ifOutNUcastPkts object and is not counted by the
                corresponding instance of the
                dot3StatsSingleCollisionFrames object."
        ::= { dot3StatsEntry 5 }

    dot3StatsSQETestErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        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 message is defined
                in section 7.2.2.2.4 of [12] and its generation is
                described in section 7.2.4.6 of the same
                document."
        ::= { dot3StatsEntry 6 }

    dot3StatsDeferredTransmissions OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        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."
        ::= { dot3StatsEntry 7 }

    dot3StatsLateCollisions OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
```

```
        DESCRIPTION
                "The number of times that a collision is detected
                on a particular interface later than 512 bit-times
                into the transmission of a packet.

                Five hundred and twelve bit-times corresponds to
                51.2 microseconds on a 10 Mbit/s system. 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."
        ::= { dot3StatsEntry 8 }

    dot3StatsExcessiveCollisions OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A count of frames for which transmission on a
                particular interface fails due to excessive
                collisions."
        ::= { dot3StatsEntry 9 }

    dot3StatsInternalMacTransmitErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        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, the
                dot3StatsCarrierSenseErrors object, or the
                dot3StatsExcessiveDeferrals 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."
        ::= { dot3StatsEntry 10 }
```

```
    dot3StatsCarrierSenseErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        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."
        ::= { dot3StatsEntry 11 }

    dot3StatsExcessiveDeferrals OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A count of frames for which transmission on a
                particular interface is deferred for an excessive
                period of time."
        ::= { dot3StatsEntry 12 }

    dot3StatsFrameTooLongs OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        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 [9], counted exclusively
                according to the error status presented to the
                LLC."
        ::= { dot3StatsEntry 13 }
```

```
    dot3StatsInRangeLengthErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A count of frames received on a particular
                interface with a length field value that falls
                between the minimum unpadded LLC data size and the
                maximum allowed LLC data size inclusive and that
                does not match the number of LLC data octets
                received.

                The count represented by an instance of this
                object also includes frames for which the length
                field value is less than the minimum unpadded LLC
                data size."
        ::= { dot3StatsEntry 14 }

    dot3StatsOutOfRangeLengthFields OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A count of frames received on a particular
                interface for which the length field value exceeds
                the maximum allowed LLC data size.

                The count represented by an instance of this
                object is not incremented in implementations that
                observe Ethernet encapsulation conventions (by
                which the IEEE 802.3 length field is interpreted
                as the Ethernet Type field)."
        ::= { dot3StatsEntry 15 }

    dot3StatsInternalMacReceiveErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        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, the
                dot3StatsFCSErrors object, the
                dot3StatsInRangeLengthErrors object, or the
```

```
                dot3StatsOutOfRangeLengthFields 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."
        ::= { dot3StatsEntry 16 }


    -- 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
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "A collection of collision histograms for a
                particular set of interfaces."
        ::= { dot3 5 }

    dot3CollEntry OBJECT-TYPE
        SYNTAX  Dot3CollEntry
        ACCESS  not-accessible
        STATUS  mandatory
        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   { dot3CollIndex, dot3CollCount }
        ::= { dot3CollTable 1 }

    Dot3CollEntry ::=
        SEQUENCE {
            dot3CollIndex
                INTEGER,
            dot3CollCount
                INTEGER,
            dot3CollFrequencies
                Counter
```

```
        }

    dot3CollIndex OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The index value that uniquely identifies the
                interface to which a particular collision
                histogram cell pertains.  The interface identified
                by a particular value of this index is the same
                interface as identified by the same value of
                ifIndex."
        ::= { dot3CollEntry 1 }

    dot3CollCount OBJECT-TYPE
        SYNTAX  INTEGER (1..16)
        ACCESS  read-only
        STATUS  mandatory
        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  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A count of individual MAC frames for which the
                transmission (successful or otherwise) on a
                particular interface is accompanied by a
                particular number of media collisions."
        ::= { dot3CollEntry 3 }


    --  802.3 Tests

    -- The ifExtnsTestTable defined in [11] provides a common means
    -- for a manager to test any interface corresponding to a value
    -- of ifIndex.

    -- At this time, one well known test (testFullDuplexLoopBack) is
    -- defined in [11].  For ethernet-like interfaces, this test
    -- configures the MAC chip and executes an internal loopback
    -- test of memory 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 object ifExtnsTestResult
    -- (defined in [11]) will be set to failed(7).  The following two
    -- OBJECT IDENTIFIERs may be used to provided more information as
    -- values for the object ifExtnsTestCode in [11]:

    dot3Errors             OBJECT IDENTIFIER ::= { dot3 7 }

                              -- couldn't initialize MAC chip for test
    dot3ErrorInitError     OBJECT IDENTIFIER ::= { dot3Errors 1 }

                              -- expected data not received (or not
                              -- received correctly) in loopback test
    dot3ErrorLoopbackError OBJECT IDENTIFIER ::= { dot3Errors 2 }


    --  TDR Test

    -- Another test, specific to ethernet-like interfaces,
    -- is Time-domain Reflectometry (TDR) which is defined
    -- as follows:

    dot3Tests   OBJECT IDENTIFIER ::= { dot3 6 }
    dot3TestTdr OBJECT IDENTIFIER ::= { dot3Tests 1 }

    -- A TDR test returns as its result the time interval between the
    -- most recent TDR test transmission and the subsequent detection
    -- of a collision.  This interval is based on a 10 MHz clock and
    -- should be normalized if the time base is other than 10 MHz.
    -- On successful completion of a TDR test, the result is stored
    -- as the value of the appropriate instance of the MIB object
    -- dot3TestTdrValue, and the OBJECT IDENTIFIER of that instance
    -- is stored in the corresponding instance of ifExtnsTestResult
    -- (thereby indicating where the result has been stored).


   --  802.3 Hardware Chipsets

   -- The object ifExtnsChipSet is provided in [11] to identify the
   -- MAC hardware used to communcate on an interface.  The following
   -- hardware chipsets are provided for 802.3:

   dot3ChipSets          OBJECT IDENTIFIER ::= { dot3 8 }
   dot3ChipSetAMD        OBJECT IDENTIFIER ::= { dot3ChipSets 1 }
   dot3ChipSetAMD7990    OBJECT IDENTIFIER ::= { dot3ChipSetAMD 1 }
```

```
   dot3ChipSetAMD79900   OBJECT IDENTIFIER ::= { dot3ChipSetAMD 2 }

   dot3ChipSetIntel      OBJECT IDENTIFIER ::= { dot3ChipSets 2 }
   dot3ChipSetIntel82586 OBJECT IDENTIFIER ::= { dot3ChipSetIntel 1 }
   dot3ChipSetIntel82596 OBJECT IDENTIFIER ::= { dot3ChipSetIntel 2 }

   dot3ChipSetSeeq       OBJECT IDENTIFIER ::= { dot3ChipSets 3 }
   dot3ChipSetSeeq8003   OBJECT IDENTIFIER ::= { dot3ChipSetSeeq 1 }

   dot3ChipSetNational      OBJECT IDENTIFIER ::= { dot3ChipSets 4 }
   dot3ChipSetNational8390  OBJECT IDENTIFIER ::=
                                          { dot3ChipSetNational 1 }
   dot3ChipSetNationalSonic OBJECT IDENTIFIER ::=
                                          { dot3ChipSetNational 2 }

   dot3ChipSetFujitsu       OBJECT IDENTIFIER ::= { dot3ChipSets 5 }
   dot3ChipSetFujitsu86950  OBJECT IDENTIFIER ::=
                                         { dot3ChipSetFujitsu 1 }

   -- For those chipsets not represented above, OBJECT IDENTIFIER
   -- assignment is required in other documentation, e.g., assignment
   -- within that part of the registration tree delegated to
   -- individual enterprises (see [3]).

   END
```

6. Acknowledgements

This document was produced by the Transmission MIB Working Group.

This document is based on a document written by Frank Kastenholz of Interlan entitled IEEE 802.3 Layer Management Draft M compatible MIB for TCP/IP Networks [10]. This document has been 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 its current concise format. Thanks to Frank Kastenholz of Interlan and Louis Steinberg of IBM for their experimentation.

7. References

[1] Cerf, V., "IAB Recommendations for the Development of Internet Network Management Standards", RFC 1052, NRI, April 1988.

[2] Cerf, V., "Report of the Second Ad Hoc Network Management Review

Group", RFC 1109, NRI, August 1989.

[3] Rose M., and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based internets", RFC 1155, Performance Systems International, Hughes LAN Systems, May 1990.

[4] McCloghrie K., and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets", RFC 1156, Hughes LAN Systems, Performance Systems International, May 1990.

[5] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol", RFC 1157, SNMP Research, Performance Systems International, Performance Systems International, MIT Laboratory for Computer Science, May 1990.

[6] McCloghrie K., and M. Rose, Editors, "Management Information Base for Network Management of TCP/IP-based internets", RFC 1213, Performance Systems International, March 1991.

[7] Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1), International Organization for Standardization, International Standard 8824, December 1987.

[8] Information processing systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Notation One (ASN.1), International Organization for Standardization, International Standard 8825, December 1987.

[9] IEEE, "IEEE 802.3 Layer Management", November 1988.

[10] Kastenholz, F., IEEE 802.3 Layer Management Draft compatible MIB for TCP/IP Networks, electronic mail message to mib-wg@nnsc.nsf.net, 9 June 1989.

[11] McCloghrie, K., Editor, "Extensions to the Generic-Interface MIB", RFC 1229, Hughes LAN Systems, Inc., May 1991.

[12] IEEE, "Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications", ANSI/IEEE Std 802.3-1985.

[13] Rose, M., and K. McCloghrie, Editors, "Concise MIB Definitions", RFC 1212, Performance Systems International, Hughes LAN Systems, March 1991.

Security Considerations

Security issues are not discussed in this memo.

Author's Address

John Cook
Chipcom Corporation
118 Turnpike Road
Southborough, MA 01772

For more information, contact the chair of the Ethernet MIB working group:

Frank Kastenholz
Clearpoint Research Inc
35 Parkwood Drive
Hopkinton Mass 01748

Phone: 508-435-2000
EMail: kasten@europa.clearpoint.com