

Network Working Group  
Request for Comments: 1231

K. McCloghrie  
Hughes LAN Systems, Inc.  
R. Fox  
Synoptics, Inc.  
E. Decker  
cisco Systems, Inc.  
May 1991

## IEEE 802.5 Token Ring MIB

### Status of this Memo

This memo defines a MIB for 805.5 networks for use with the SNMP protocol. This memo is a product of the Transmission Working Group of the Internet Engineering Task Force (IETF). 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 .....	1
2. The Network Management Framework.....	2
3. Objects .....	2
3.1 Format of Definitions .....	3
4. Overview .....	3
4.1 Scope of Definitions .....	3
4.2 Textual Conventions .....	3
5. Definitions .....	4
6. Acknowledgements .....	21
7. References .....	22
8. Security Considerations.....	23
9. Authors' Addresses.....	23

### 1. Abstract

This memo defines an experimental portion of the Management Information Base (MIB) for use with network management protocols in TCP/IP-based internets. In particular, this memo defines managed objects used for managing subnetworks which use the IEEE 802.5 Token Ring technology described in 802.5 Token Ring Access Method and Physical Layer Specifications, IEEE Standard 802.5-1989.

## 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 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 [9,10].

## 4. Overview

This memo defines three tables: the 802.5 Interface Table, which contains state and parameter information which is specific to 802.5 interfaces, the 802.5 Statistics Table, which contains 802.5 interface statistics, and the 802.5 Timer Table, which contains the values of 802.5-defined timers. A managed system will have one entry in the 802.5 Interface Table and one entry in the 802.5 Statistics Table for each of its 802.5 interfaces. Implementation of the 802.5 Timer Table is optional.

This memo also defines OBJECT IDENTIFIERS, some to identify 802.5 tests, for use with the ifExtnsTestTable defined in [11], and some to identify Token Ring interface Chip Sets, for use with the ifExtnsChipSet object defined in [11].

### 4.1. Scope of Definitions

All objects defined in this memo are registered in a single subtree within the experimental namespace [3], and are for use with every interface which conforms to the IEEE 802.5 Token Ring Access Method and Physical Layer Specifications [10]. At present, this applies to interfaces for which the ifType variable in the Internet-standard MIB [4,6] has the value:

iso88025-tokenRing(9)

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

dot5      OBJECT IDENTIFIER ::= { experimental 4 }

as defined below.

### 4.2. Textual Conventions

A new datatype, MacAddress, is introduced as a textual convention in this document. This textual convention has NO effect on either the syntax nor the semantics of any managed object. Objects defined using this convention are always encoded by means of the rules that define their primitive type. Hence, no changes to the SMI or the SNMP are

necessary to accommodate this textual convention which is adopted merely for the convenience of readers.

## 5. Definitions

```
RFC1231-MIB DEFINITIONS ::= BEGIN
```

```
--                               IEEE 802.5 Token Ring MIB
```

```
IMPORTS
```

```
    experimental
        FROM RFC1155-SMI
    OBJECT-TYPE
        FROM RFC-1212;
```

```
-- This MIB Module uses the extended OBJECT-TYPE macro as
-- defined in [9].
```

```
dot5    OBJECT IDENTIFIER ::= { experimental 4 }
```

```
-- All representations of MAC addresses in this MIB Module
-- use, as a textual convention (i.e. this convention does
-- not affect their encoding), the data type:
```

```
MacAddress ::= OCTET STRING (SIZE (6))    -- a 6 octet
                                           -- address in the
                                           -- "canonical" order
-- defined by IEEE 802.1a, i.e., as if it were transmitted
-- least significant bit first, even though 802.5 (in
-- contrast to other 802.x protocols) requires MAC addresses
-- to be transmitted most significant bit first.
--
-- 16-bit addresses, if needed, are represented by setting
-- their upper 4 octets to all 0's, i.e., AAFF would be
-- represented as 00000000AAFF.
```

```
-- The Interface Table
```

```
-- This table contains state and parameter information which
-- is specific to 802.5 interfaces. It is mandatory that
-- systems having 802.5 interfaces implement this table in
-- addition to the generic interfaces table [4,6] and its
-- generic extensions [11].
```

```
dot5Table OBJECT-TYPE
SYNTAX SEQUENCE OF Dot5Entry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "This table contains Token Ring interface
    parameters and state variables, one entry
    per 802.5 interface."
 ::= { dot5 1 }

dot5Entry OBJECT-TYPE
SYNTAX Dot5Entry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "A list of Token Ring status and parameter
    values for an 802.5 interface."
INDEX { dot5IfIndex }
 ::= { dot5Table 1 }

Dot5Entry
 ::= SEQUENCE {
     dot5IfIndex
         INTEGER,
     dot5Commands
         INTEGER,
     dot5RingStatus
         INTEGER,
     dot5RingState
         INTEGER,
     dot5RingOpenStatus
         INTEGER,
     dot5RingSpeed
         INTEGER,
     dot5UpStream
         MacAddress,
     dot5ActMonParticipate
         INTEGER,
     dot5Functional
         MacAddress
 }

dot5IfIndex OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The value of this object identifies the
```

802.5 interface for which this entry contains management information. The value of this object for a particular interface has the same value as the ifIndex object, defined in [4,6], for the same interface."

::= { dot5Entry 1 }

dot5Commands OBJECT-TYPE

SYNTAX INTEGER {  
no-op(1),  
open(2),  
reset(3),  
close(4)  
}

ACCESS read-write

STATUS mandatory

DESCRIPTION

"When this object is set to the value of open(2), the station should go into the open state. The progress and success of the open is given by the values of the objects dot5RingState and dot5RingOpenStatus.

When this object is set to the value of reset(3), then the station should do a reset. On a reset, all MIB counters should retain their values, if possible. Other side affects are dependent on the hardware chip set.

When this object is set to the value of close(4), the station should go into the stopped state by removing itself from the ring.

Setting this object to a value of no-op(1) has no effect.

When read, this object always has a value of no-op(1)."

::= { dot5Entry 2 }

dot5RingStatus OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The current interface status which can be used to diagnose fluctuating problems that can occur on token rings, after a

station has successfully been added to the ring.

Before an open is completed, this object has the value for the 'no status' condition. The dot5RingState and dot5RingOpenStatus objects provide for debugging problems when the station can not even enter the ring.

The object's value is a sum of values, one for each currently applicable condition. The following values are defined for various conditions:

```

        0 = No Problems detected
        32 = Ring Recovery
        64 = Single Station
        256 = Remove Received
        512 = reserved
        1024 = Auto-Removal Error
        2048 = Lobe Wire Fault
        4096 = Transmit Beacon
        8192 = Soft Error
        16384 = Hard Error
        32768 = Signal Loss
        131072 = no status, open not completed."
 ::= { dot5Entry 3 }

```

dot5RingState OBJECT-TYPE

```

SYNTAX INTEGER {
    opened(1),
    closed(2),
    opening(3),
    closing(4),
    openFailure(5),
    ringFailure(6)
}

```

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The current interface state with respect to entering or leaving the ring."

```
 ::= { dot5Entry 4 }
```

dot5RingOpenStatus OBJECT-TYPE

```

SYNTAX INTEGER {
    noOpen(1),           -- no open attempted
    badParam(2),
    lobeFailed(3),
}

```

```

        signalLoss(4),
        insertionTimeout(5),
        ringFailed(6),
        beaconing(7),
        duplicateMAC(8),
        requestFailed(9),
        removeReceived(10),
        open(11)      -- last open successful
    }
dot5RingStatus OBJECT-TYPE
    SYNTAX      INTEGER {
        unknown(1),
        oneMegabit(2),
        fourMegabit(3),
        sixteenMegabit(4)
    }
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "This object indicates the success, or the
        reason for failure, of the station's most
        recent attempt to enter the ring."
    ::= { dot5Entry 5 }

dot5RingSpeed OBJECT-TYPE
    SYNTAX      INTEGER {
        unknown(1),
        oneMegabit(2),
        fourMegabit(3),
        sixteenMegabit(4)
    }
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "The ring's bandwidth."
    ::= { dot5Entry 6 }

dot5UpStream OBJECT-TYPE
    SYNTAX      MacAddress
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "The MAC-address of the up stream neighbor
        station in the ring."
    ::= { dot5Entry 7 }

dot5ActMonParticipate OBJECT-TYPE
    SYNTAX      INTEGER {
        true(1),
        false(2)
    }
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "If this object has a value of true(1) then

```



this interface will participate in the active monitor selection process. If the value is false(2) then it will not. Setting this object might not have an effect until the next time the interface is opened."

::= { dot5Entry 8 }

dot5Functional OBJECT-TYPE

SYNTAX MacAddress

ACCESS read-write

STATUS mandatory

DESCRIPTION

"The bit mask of all Token Ring functional addresses for which this interface will accept frames."

::= { dot5Entry 9 }

-- The Statistics Table

-- This table contains statistics and error counter which are  
-- specific to 802.5 interfaces. It is mandatory that systems  
-- having 802.5 interfaces implement this table.

dot5StatsTable OBJECT-TYPE

SYNTAX SEQUENCE OF Dot5StatsEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"A table containing Token Ring statistics, one entry per 802.5 interface.

All the statistics are defined using the syntax Counter as 32-bit wrap around counters. Thus, if an interface's hardware maintains these statistics in 16-bit counters, then the agent must read the hardware's counters frequently enough to prevent loss of significance, in order to maintain 32-bit counters in software."

::= { dot5 2 }

dot5StatsEntry OBJECT-TYPE

SYNTAX Dot5StatsEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"An entry contains the 802.5 statistics  
for a particular interface."  
INDEX { dot5StatsIfIndex }  
 ::= { dot5StatsTable 1 }

Dot5StatsEntry  
 ::= SEQUENCE {  
 dot5StatsIfIndex  
 INTEGER,  
 dot5StatsLineErrors  
 Counter,  
 dot5StatsBurstErrors  
 Counter,  
 dot5StatsACErrors  
 Counter,  
 dot5StatsAbortTransErrors  
 Counter,  
 dot5StatsInternalErrors  
 Counter,  
 dot5StatsLostFrameErrors  
 Counter,  
 dot5StatsReceiveCongestions  
 Counter,  
 dot5StatsFrameCopiedErrors  
 Counter,  
 dot5StatsTokenErrors  
 Counter,  
 dot5StatsSoftErrors  
 Counter,  
 dot5StatsHardErrors  
 Counter,  
 dot5StatsSignalLoss  
 Counter,  
 dot5StatsTransmitBeacons  
 Counter,  
 dot5StatsRecoverys  
 Counter,  
 dot5StatsLobeWires  
 Counter,  
 dot5StatsRemoves  
 Counter,  
 dot5StatsSingles  
 Counter,  
 dot5StatsFreqErrors  
 Counter  
 }

```
dot5StatsIfIndex OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The value of this object identifies the
        802.5 interface for which this entry
        contains management information. The
        value of this object for a particular
        interface has the same value as the
        ifIndex object, defined in [4,6], for
        the same interface."
    ::= { dot5StatsEntry 1 }

dot5StatsLineErrors OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "This counter is incremented when a frame
        or token is copied or repeated by a
        station, the E bit is zero in the frame
        or token and one of the following
        conditions exists: 1) there is a
        non-data bit (J or K bit) between the SD
        and the ED of the frame or token, or
        2) there is an FCS error in the frame."
    ::= { dot5StatsEntry 2 }

dot5StatsBurstErrors OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "This counter is incremented when a station
        detects the absence of transitions for five
        half-bit timers (burst-five error)."
    ::= { dot5StatsEntry 3 }

dot5StatsACErrors OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "This counter is incremented when a station
        receives an AMP or SMP frame in which A is
        equal to C is equal to 0, and then receives
        another SMP frame with A is equal to C is
```

equal to 0 without first receiving an AMP frame. It denotes a station that cannot set the AC bits properly."  
 ::= { dot5StatsEntry 4 }

dot5StatsAbortTransErrors OBJECT-TYPE

SYNTAX Counter  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"This counter is incremented when a station transmits an abort delimiter while transmitting."  
 ::= { dot5StatsEntry 5 }

dot5StatsInternalErrors OBJECT-TYPE

SYNTAX Counter  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"This counter is incremented when a station recognizes an internal error."  
 ::= { dot5StatsEntry 6 }

dot5StatsLostFrameErrors OBJECT-TYPE

SYNTAX Counter  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"This counter is incremented when a station is transmitting and its TRR timer expires. This condition denotes a condition where a transmitting station in strip mode does not receive the trailer of the frame before the TRR timer goes off."  
 ::= { dot5StatsEntry 7 }

dot5StatsReceiveCongestions OBJECT-TYPE

SYNTAX Counter  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"This counter is incremented when a station recognizes a frame addressed to its specific address, but has no available buffer space indicating that the station is congested."  
 ::= { dot5StatsEntry 8 }

**dot5StatsFrameCopiedErrors OBJECT-TYPE**

**SYNTAX** Counter  
**ACCESS** read-only  
**STATUS** mandatory  
**DESCRIPTION**

"This counter is incremented when a station recognizes a frame addressed to its specific address and detects that the FS field A bits are set to 1 indicating a possible line hit or duplicate address."

::= { dot5StatsEntry 9 }

**dot5StatsTokenErrors OBJECT-TYPE**

**SYNTAX** Counter  
**ACCESS** read-only  
**STATUS** mandatory  
**DESCRIPTION**

"This counter is incremented when a station acting as the active monitor recognizes an error condition that needs a token transmitted."

::= { dot5StatsEntry 10 }

**dot5StatsSoftErrors OBJECT-TYPE**

**SYNTAX** Counter  
**ACCESS** read-only  
**STATUS** mandatory  
**DESCRIPTION**

"The number of Soft Errors the interface has detected. It directly corresponds to the number of Report Error MAC frames that this interface has transmitted. Soft Errors are those which are recoverable by the MAC layer protocols."

::= { dot5StatsEntry 11 }

**dot5StatsHardErrors OBJECT-TYPE**

**SYNTAX** Counter  
**ACCESS** read-only  
**STATUS** mandatory  
**DESCRIPTION**

"The number of times this interface has detected an immediately recoverable fatal error. It denotes the number of times this interface is either transmitting or receiving beacon MAC frames."

::= { dot5StatsEntry 12 }

```
dot5StatsSignalLoss OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of times this interface has
        detected the loss of signal condition from
        the ring."
    ::= { dot5StatsEntry 13 }

dot5StatsTransmitBeacons OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of times this interface has
        transmitted a beacon frame."
    ::= { dot5StatsEntry 14 }

dot5StatsRecoverys OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of Claim Token MAC frames
        received or transmitted after the interface
        has received a Ring Purge MAC frame. This
        counter signifies the number of times the
        ring has been purged and is being recovered
        back into a normal operating state."
    ::= { dot5StatsEntry 15 }

dot5StatsLobeWires OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of times the interface has
        detected an open or short circuit in the
        lobe data path. The adapter will be closed
        and dot5RingState will signify this
        condition."
    ::= { dot5StatsEntry 16 }

dot5StatsRemoves OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
```

## DESCRIPTION

"The number of times the interface has received a Remove Ring Station MAC frame request. When this frame is received the interface will enter the close state and dot5RingState will signify this condition."

::= { dot5StatsEntry 17 }

## dot5StatsSingles OBJECT-TYPE

SYNTAX Counter

ACCESS read-only

STATUS mandatory

## DESCRIPTION

"The number of times the interface has sensed that it is the only station on the ring. This will happen if the interface is the first one up on a ring, or if there is a hardware problem."

::= { dot5StatsEntry 18 }

## dot5StatsFreqErrors OBJECT-TYPE

SYNTAX Counter

ACCESS read-only

STATUS optional

## DESCRIPTION

"The number of times the interface has detected that the frequency of the incoming signal differs from the expected frequency by more than that specified by the IEEE 802.5 standard, see chapter 7 in [10]."

::= { dot5StatsEntry 19 }

-- The Timer Table

-- This group contains the values of the timers defined in  
-- [10] for 802.5 interfaces. It is optional that systems  
-- having 802.5 interfaces implement this group.

## dot5TimerTable OBJECT-TYPE

SYNTAX SEQUENCE OF Dot5TimerEntry

ACCESS not-accessible

STATUS mandatory

## DESCRIPTION

"This table contains Token Ring interface timer values, one entry per 802.5

```

        interface."
 ::= { dot5 5 }

dot5TimerEntry OBJECT-TYPE
    SYNTAX Dot5TimerEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "A list of Token Ring timer values for an
        802.5 interface."
    INDEX { dot5TimerIfIndex }
    ::= { dot5TimerTable 1 }

Dot5TimerEntry
 ::= SEQUENCE {
     dot5TimerIfIndex
         INTEGER,
     dot5TimerReturnRepeat
         INTEGER,
     dot5TimerHolding
         INTEGER,
     dot5TimerQueuePDU
         INTEGER,
     dot5TimerValidTransmit
         INTEGER,
     dot5TimerNoToken
         INTEGER,
     dot5TimerActiveMon
         INTEGER,
     dot5TimerStandbyMon
         INTEGER,
     dot5TimerErrorReport
         INTEGER,
     dot5TimerBeaconTransmit
         INTEGER,
     dot5TimerBeaconReceive
         INTEGER
 }

dot5TimerIfIndex OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The value of this object identifies the
        802.5 interface for which this entry
        contains timer values. The value of
        this object for a particular interface

```



has the same value as the ifIndex object, defined in [4,6], for the same interface."

::= { dot5TimerEntry 1 }

dot5TimerReturnRepeat OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The time-out value used to ensure the interface will return to Repeat State, in units of 100 micro-seconds. The value should be greater than the maximum ring latency.

Implementors are encouraged to provide read-write access to this object if that is possible/useful in their system, but giving due consideration to the dangers of write-able timers."

::= { dot5TimerEntry 2 }

dot5TimerHolding OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Maximum period of time a station is permitted to transmit frames after capturing a token, in units of 100 micro-seconds.

Implementors are encouraged to provide read-write access to this object if that is possible/useful in their system, but giving due consideration to the dangers of write-able timers."

::= { dot5TimerEntry 3 }

dot5TimerQueuePDU OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The time-out value for enqueueing of an SMP PDU after reception of an AMP or SMP frame in which the A and C bits were equal to 0, in units of 100 micro-seconds.

Implementors are encouraged to provide

read-write access to this object if that is possible/useful in their system, but giving due consideration to the dangers of write-able timers."

::= { dot5TimerEntry 4 }

dot5TimerValidTransmit OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The time-out value used by the active monitor to detect the absence of valid transmissions, in units of 100 micro-seconds.

Implementors are encouraged to provide read-write access to this object if that is possible/useful in their system, but giving due consideration to the dangers of write-able timers."

::= { dot5TimerEntry 5 }

dot5TimerNoToken OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The time-out value used to recover from various-related error situations [9]. If N is the maximum number of stations on the ring, the value of this timer is normally:

dot5TimerReturnRepeat + N\*dot5TimerHolding.

Implementors are encouraged to provide read-write access to this object if that is possible/useful in their system, but giving due consideration to the dangers of write-able timers."

::= { dot5TimerEntry 6 }

dot5TimerActiveMon OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The time-out value used by the active monitor to stimulate the enqueueing of an AMP PDU for transmission, in units of

100 micro-seconds.

Implementors are encouraged to provide read-write access to this object if that is possible/useful in their system, but giving due consideration to the dangers of write-able timers."

::= { dot5TimerEntry 7 }

dot5TimerStandbyMon OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The time-out value used by the stand-by monitors to ensure that there is an active monitor on the ring and to detect a continuous stream of tokens, in units of 100 micro-seconds.

Implementors are encouraged to provide read-write access to this object if that is possible/useful in their system, but giving due consideration to the dangers of write-able timers."

::= { dot5TimerEntry 8 }

dot5TimerErrorReport OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The time-out value which determines how often a station shall send a Report Error MAC frame to report its error counters, in units of 100 micro-seconds.

Implementors are encouraged to provide read-write access to this object if that is possible/useful in their system, but giving due consideration to the dangers of write-able timers."

::= { dot5TimerEntry 9 }

dot5TimerBeaconTransmit OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The time-out value which determines how long a station shall remain in the state

of transmitting Beacon frames before entering the Bypass state, in units of 100 micro-seconds.

Implementors are encouraged to provide read-write access to this object if that is possible/useful in their system, but giving due consideration to the dangers of write-able timers."

::= { dot5TimerEntry 10 }

dot5TimerBeaconReceive OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The time-out value which determines how long a station shall receive Beacon frames from its downstream neighbor before entering the Bypass state, in units of 100 micro-seconds.

Implementors are encouraged to provide read-write access to this object if that is possible/useful in their system, but giving due consideration to the dangers of write-able timers."

::= { dot5TimerEntry 11 }

--

## 802.5 Interface Tests

dot5Tests OBJECT IDENTIFIER ::= { dot5 3 }

-- The extensions to the interfaces table proposed in [11]  
 -- define a table object, ifExtnsTestTable, through which a  
 -- network manager can instruct an agent to test an interface  
 -- for various faults. A test to be performed is identified  
 -- (as the value of ifExtnsTestType) via an OBJECT IDENTIFIER.

--

-- The Full-Duplex Loop Back Test is a common test, defined  
 -- in [11] as:

--

-- testFullDuplexLoopBack

--

-- Invoking this test on a 802.5 interface causes the  
 -- interface to check the path from memory through the  
 -- chip set's internal logic and back to memory, thus  
 -- checking the proper functioning of the systems's  
 -- interface to the chip set.

-- The Insert Function test is defined by:

testInsertFunc      OBJECT IDENTIFIER ::= { dot5Tests 1 }

-- Invoking this test causes the station to test the insert  
-- ring logic of the hardware if the station's lobe media  
-- cable is connected to a wiring concentrator. Note that  
-- this command inserts the station into the network, and  
-- thus, could cause problems if the station is connected  
-- to a operational network.

--                              802.5 Hardware Chip Sets

dot5ChipSets      OBJECT IDENTIFIER ::= { dot5 4 }

-- The extensions to the interfaces table proposed in [11]  
-- define an object, ifExtnsChipSet, with the syntax of  
-- OBJECT IDENTIFIER, to identify the hardware chip set in  
-- use by an interface. That definition specifies just  
-- one applicable object identifier:

--              unknownChipSet

-- for use as the value of ifExtnsChipSet when the specific  
-- chip set is unknown.

-- This MIB defines the following for use as values of  
-- ifExtnsChipSet:

-- IBM 16/4 Mb/s

chipSetIBM16      OBJECT IDENTIFIER ::= { dot5ChipSets 1 }

-- TI 4Mb/s

chipSetTIitms380      OBJECT IDENTIFIER ::= { dot5ChipSets 2 }

-- TI 16/4 Mb/s

chipSetTIitms380c16 OBJECT IDENTIFIER ::= { dot5ChipSets 3 }

END

## 6. Acknowledgements

This document was produced under the auspices of the IETF's  
Transmission Working Group. The comments of the following  
individuals are acknowledged:

Tom Benkart, Advanced Computer Communications  
Stan Froyd, Advanced Computer Communications  
Marshall T. Rose, Performance Systems International, Inc.

## 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 (SNMP)", 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] Rose, M., and K. McCloghrie, Editors, "Concise MIB Definitions", RFC 1212, Performance Systems International, Hughes LAN Systems, March 1991.
- [10] Token Ring Access Method and Physical Layer Specifications, Institute of Electrical and Electronic Engineers, IEEE Standard 802.5-1989, 1989.

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

## 8. Security Considerations

Security issues are not discussed in this memo.

## 9. Authors' Addresses

Keith McCloghrie  
Hughes LAN Systems, Inc.  
1225 Charleston Road  
Mountain View, CA 94043

Phone: (415) 966-7934  
EMail: kzm@hls.com

Richard Fox  
Synoptics, Inc.  
4401 Great America Pkwy  
PO Box 58185  
Santa Clara, Cal. 95052

Phone: (408) 764-1372  
EMail: rfox@synoptics.com

Eric Decker  
cisco Systems, Inc.  
1525 O'Brien Dr.  
Menlo Park, CA 94025

Phone: (415) 688-8241  
EMail: cire@cisco.com