

## SNMP MIB Extension for X.25 LAPB

### Status of this Memo

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.

### 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 the Link Layer of X.25, LAPB. The objects defined here, along with the objects in the "SNMP MIB Extension for the Packet Layer of X.25" [9] and the "Definitions of Managed Objects for RS-232-like Hardware Devices" [8], combine to allow management of an X.25 protocol stack.

### Table of Contents

|   |    |
|---|----|
| 1. The Network Management Framework ..... | 2  |
| 2. Objects .....                          | 2  |
| 2.1 Format of Definitions .....           | 3  |
| 3. Overview .....                         | 3  |
| 3.1 Informal overview .....               | 3  |
| 3.2 Textual Conventions .....             | 4  |
| 3.3 Formal overview .....                 | 4  |
| 3.4 Tables .....                          | 5  |
| 3.5 Traps .....                           | 6  |
| 4. Object Definitions .....               | 6  |
| 5. Appendix: Revision History .....       | 27 |
| July 30, 1992 .....                       | 27 |
| June 12, 1992 .....                       | 27 |
| May 18, 1992 .....                        | 28 |
| April 8, 1992 .....                       | 28 |
| February 1992 .....                       | 28 |
| October 1991 .....                        | 29 |
| June 1991 .....                           | 30 |
| April 1991 .....                          | 30 |

|                                  |    |
|----------------------------------|----|
| 6. Acknowledgements .....        | 30 |
| 7. References .....              | 31 |
| 8. Security Considerations ..... | 33 |
| 9. Authors' Addresses .....      | 33 |

## 1. The Network Management Framework

The Internet-standard Network Management Framework consists of three components. These components give the rules for defining objects, the definitions of objects, and the protocol for manipulating objects.

The network management framework structures objects in an abstract information tree. The branches of the tree name objects and the leaves of the tree contain the values manipulated to effect management. This tree is called the Management Information Base or MIB. The concepts of this tree are given in STD 16/RFC 1155 "The Structure of Management Information" or SMI [1]. The SMI defines the trunk of the tree and the types of objects used when defining the leaves. STD 16/RFC 1212, "Towards Concise MIB Definitions" [4], defines a more concise description mechanism that preserves all the principals of the SMI.

The core MIB definitions for the Internet suite of protocols can be found in RFC 1156 [2] "Management Information Base for Network Management of TCP/IP-based internets". STD 17/RFC 1213 [5] defines MIB-II, an evolution of MIB-I with changes to incorporate implementation experience and new operational requirements.

STD 15/RFC 1157 [3] defines the SNMP protocol itself. The protocol defines how to manipulate the objects in a remote MIB.

The tree structure of the MIB allows new objects to be defined for the purpose of experimentation and evaluation.

## 2. Objects

The definition of an object in the MIB requires an object name and type. Object names and types are defined using the subset of the Abstract Syntax Notation One (ASN.1) [6] defined in the SMI [1]. Objects are named using ASN.1 object identifiers, administratively assigned names, to specify object types. The object name, together with an optional object instance, uniquely identifies a specific instance of an object. For human convenience, we often use a textual string, termed the OBJECT DESCRIPTOR, to also refer to objects.

Objects also have a syntax that defines the abstract data structure corresponding to that object type. The ASN.1 language [6] provides

the primitives used for this purpose. The SMI [1] purposely restricts the ASN.1 constructs which may be used for simplicity and ease of implementation. The encoding of an object type simply describes how to represent an object using ASN.1 encoding rules [7], for purposes of dealing with the SNMP protocol.

## 2.1. Format of Definitions

Section 4 contains the specification of all object types defined in this MIB module. The object definitions use the conventions given in the SMI [1] as amended by the concise MIB definitions [4].

## 3. Overview

### 3.1. Informal overview

This section describes how the objects defined below relate with other MIBs. This section is only informational to help understand how the pieces fit together.

The objects defined below are to be used in conjunction with MIB-II and other MIBs such as the X.25 packet level MIB [9]. A system with a complete X.25 stack running over a synchronous line will have at least two interfaces in the ifTable defined in MIB-II. There will be an interface for LAPB and another interface for the packet layer of X.25. There will also be objects defined in the RS-232-like MIB for the physical sync line.

Each software interface identifies the layer below it used to send and receive packets. The X.25 MIB object, x25InfoDataLinkId, specifies an instance of lapbAdmnIndex for the LAPB interface under that X.25. The LAPB object, lapbOperPortId, defined below, identifies an instance of the rs232PortIndex for the the Sync line used by LAPB.

For X.25 running over LAPB over Ethernet, the lapbAdmnPortId would identify the instance of ifIndex for the Ethernet interface.

Each X.25 subnetwork will have separate entries in the ifTable. Thus a system with two X.25 lines would have two ifTable entries for the two X.25 packet layers and two other entries for the two LAPB interfaces. Each X.25 Packet Layer MIB would identify the instance of the LAPB MIB below it. Each LAPB MIB would identify the Sync line below it. The system would also have two entries for rs232PortTable and rs232SyncPortTable for the two physical lines.

Since the ifTable as defined in MIB-II is device independent, it doesn't have anything specific for any type of interface. The objects below define the LAPB specific information for an interface

of type LAPB. Different LAPB interfaces can also be differentiated by matching the values of ifIndex with lapbAdmnIndex.

### 3.2. Textual Conventions

Two new data types are introduced as a textual conventions in this MIB document. These textual conventions enhance the readability of the specification and can ease comparison with other specifications if appropriate. It should be noted that the introduction of these textual conventions has no effect on either the syntax nor the semantics of any managed objects. The use of these is merely an artifact of the explanatory method used. Objects defined in terms of one of these methods are always encoded by means of the rules that define the primitive type. Hence, no changes to the SMI or the SNMP are necessary to accommodate these textual conventions which are adopted merely for the convenience of readers and writers in pursuit of the elusive goal of clear, concise, and unambiguous MIB documents.

This MIB introduces the data types of:

PositiveInteger  
ifIndexType

### 3.3. Formal overview

Instances of the objects defined below represent attributes of a LAPB interface. LAPB interfaces are identified by an ifType object in the Internet-standard MIB [5] of

lapb(16).

For these interfaces, the value of the ifSpecific variable in the MIB-II [5] has the OBJECT IDENTIFIER value:

lapb      OBJECT IDENTIFIER ::= { transmission 16 }

The relationship between a LAPB interface and an interface in the context of the Internet-standard MIB [5] 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 below.

The objects defined below are defined in the context of ISO 7776 [10] and ISO 8885 [11]. Access to those documents maybe useful (but isn't essential) to understand the names and semantics of some objects. Where possible the object descriptions use the terminology of ISO 7776; for example, one commonly used term refers to the peer LAPB as the DCE/remote DTE. This terminology does not restrict the instrumented LAPB to function only as a DTE. This MIB maybe applied

to a LAPB configured as either a DCE or a DTE.

To the extent that some attributes defined in the Internet standard MIB [5] are applicable to LAPB, those objects have not been duplicated here. In some instances some clarification of how to apply those objects to LAPB has been given.

Some objects defined below include a DEFVAL clause. This clause provides reasonable (but not mandatory) default values to use when creating these objects. This does not imply this MIB defines any mechanism for creating or deleting LAPB interfaces. The creation and deletion of the objects of this MIB depend on the implementation method for creating and deleting LAPB interfaces. The DEFVAL clause provides reasonable defaults to allow further extension of the MIB to define methods for creating and deleting LAPB interfaces without having to deprecate these objects for the lack of a DEFVAL clause.

### 3.4. Tables

This extension adds four tables to the MIB. These tables are:

lapbAdmnTable,  
lapbOperTable,  
lapbFlowTable, and  
lapbXidTable.

The lapbAdmnTable provides objects for common parameters used by LAPB such as the T1 retransmission timer or the N2 retransmission counter. Changes to objects in this table need not affect a running interface but provides access to the values used to initialize an interface. These values are read-write.

The lapbOperTable provides objects to determine the parameters actually in use by an interface. These objects are read only. The values currently in use maybe different from the lapbAdmnTable values if the lapbAdmnTable was changed after interface initialization or if XID negotiation selected different values.

The lapbFlowTable provides objects that report how the LAPB interface performs. These are read-only objects used to monitor operation.

The lapbXidTable is not required for systems that do not transmit XID frames. For systems that do transmit XID frames, this table provides the values for the fields of the XID frame that are not already present in the lapbAdmnTable. The objects in this table are read-write.

### 3.5. Traps

Since all LAPB interfaces have entries in the ifTable, significant changes in the state of the interface should send a linkUp or linkDown trap. Thus an interface that receives or sends a Frame Reject frame should send a linkDown trap. If the interface later comes back up, it should then send a linkUP trap.

## 4. Object Definitions

```
RFC1381-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    Counter
        FROM RFC1155-SMI
    transmission
        FROM RFC1213-MIB
    OBJECT-TYPE
        FROM RFC-1212;
```

```
-- LAPB MIB
```

```
lapb    OBJECT IDENTIFIER ::= { transmission 16 }
```

```
PositiveInteger ::= INTEGER (0..2147483647)
```

```
IfIndexType    ::= INTEGER (1..2147483647)
```

```
-- IfIndexType specifies an index object for a table
-- with entries that match entries in the MIB-II ifTable.
-- The value of the index for the table will match the
-- ifIndex entry for same interface in the ifTable.
-- The values of this object range from 1 to ifNumber
-- inclusive.
```

```
-- #####
--                               LAPB Admn Table
-- #####
```

```
-- Support of the lapbAdmnTable is mandatory for all
-- agents of systems that implement LAPB.
```

```
lapbAdmnTable    OBJECT-TYPE
    SYNTAX    SEQUENCE OF LapbAdmnEntry
    ACCESS    not-accessible
    STATUS    mandatory
```

## DESCRIPTION

"This table contains objects that can be changed to manage a LAPB interface. Changing one of these parameters may take effect in the operating LAPB immediately or may wait until the interface is restarted depending on the details of the implementation.

Most of the objects in this read-write table have corresponding read-only objects in the lapbOperTable that return the current operating value.

The operating values may be different from these configured values if changed by XID negotiation or if a configured parameter was changed after the interface was started."

::= { lapb 1 }

```
lapbAdmnEntry    OBJECT-TYPE
    SYNTAX      LapbAdmnEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION
        "Configured parameter values for a specific
         LAPB."
    INDEX { lapbAdmnIndex }
    ::= { lapbAdmnTable 1 }
```

```
LapbAdmnEntry ::= SEQUENCE {
    lapbAdmnIndex
        IfIndexType,
    lapbAdmnStationType
        INTEGER,
    lapbAdmnControlField
        INTEGER,
    lapbAdmnTransmitNlFrameSize
        PositiveInteger,
    lapbAdmnReceiveNlFrameSize
        PositiveInteger,
    lapbAdmnTransmitKWindowSize
        INTEGER,
    lapbAdmnReceiveKWindowSize
        INTEGER,
    lapbAdmnN2RxmitCount
        INTEGER,
    lapbAdmnTlAckTimer
```

```

        PositiveInteger,
lapbAdmnT2AckDelayTimer
        PositiveInteger,
lapbAdmnT3DisconnectTimer
        PositiveInteger,
lapbAdmnT4IdleTimer
        PositiveInteger,
lapbAdmnActionInitiate
        INTEGER,
lapbAdmnActionRecvDM
        INTEGER
    }

lapbAdmnIndex    OBJECT-TYPE
    SYNTAX      IfIndexType
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "The ifIndex value for the LAPB interface."
    ::= { lapbAdmnEntry 1 }

lapbAdmnStationType    OBJECT-TYPE
    SYNTAX      INTEGER {
        dte (1),
        dce (2),
        dxe (3)
    }
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Identifies the desired station type of this
        interface."
    REFERENCE   "ISO 7776 section 3.1"
    DEFVAL      { dte }
    ::= { lapbAdmnEntry 2 }

lapbAdmnControlField OBJECT-TYPE
    SYNTAX      INTEGER {
        modulo8 (1),
        modulo128 (2)
    }
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "The desired size of the sequence numbers
        used to number frames."
    REFERENCE   "ISO 8885 Table 3, Name: HDLC Option - 10"
    DEFVAL      { modulo8 }

```



```
 ::= { lapbAdmnEntry 3 }

lapbAdmnTransmitNlFrameSize OBJECT-TYPE
    SYNTAX  PositiveInteger
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The default maximum N1 frame size desired
        in number of bits for a frame transmitted by
        this DTE. This excludes flags and 0 bits
        inserted for transparency."
    REFERENCE "ISO 8885 Table 3,
        Name: Information Field length"
    DEFVAL { 36000 } -- 4500 * 8; 802.5 Frame size
    ::= { lapbAdmnEntry 4 }

lapbAdmnReceiveNlFrameSize OBJECT-TYPE
    SYNTAX  PositiveInteger
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The default maximum N1 frame size desired
        in number of bits for a frame the DCE/remote
        DTE transmits to this DTE. This excludes
        flags and 0 bits inserted for transparency."
    DEFVAL { 36000 } -- 4500 * 8; 802.5 Frame size
    ::= { lapbAdmnEntry 5 }

lapbAdmnTransmitKWindowSize OBJECT-TYPE
    SYNTAX  INTEGER (1..127)
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The default transmit window size for this
        Interface. This is the maximum number of
        unacknowledged sequenced PDUs that may be
        outstanding from this DTE at any one time."
    REFERENCE "ISO 8885 Table 3, Name: Window size"
    DEFVAL { 7 }
    ::= { lapbAdmnEntry 6 }

lapbAdmnReceiveKWindowSize OBJECT-TYPE
    SYNTAX  INTEGER (1..127)
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The default receive window size for this
        Interface. This is the maximum number of
```

unacknowledged sequenced PDUs that may be outstanding from the DCE/remote DTE at any one time."

REFERENCE "ISO 8885 Table 3, Name: Window size"

DEFVAL { 7 }

::= { lapbAdmnEntry 7 }

lapbAdmnN2RxmitCount OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"The default N2 retry counter for this interface. This specifies the number of times a PDU will be resent after the T1 timer expires without an acknowledgement for the PDU."

REFERENCE "ISO 8885 Table 3,  
Name: Retransmission Attempts"

DEFVAL { 20 }

::= { lapbAdmnEntry 8 }

lapbAdmnT1AckTimer OBJECT-TYPE

SYNTAX PositiveInteger

ACCESS read-write

STATUS mandatory

DESCRIPTION

"The default T1 timer for this interface. This specifies the maximum time in Milliseconds to wait for acknowledgment of a PDU."

REFERENCE "ISO 8885 Table 3, Name:  
Acknowledgement timer"

DEFVAL { 3000 }

::= { lapbAdmnEntry 9 }

lapbAdmnT2AckDelayTimer OBJECT-TYPE

SYNTAX PositiveInteger

ACCESS read-write

STATUS mandatory

DESCRIPTION

"The default T2 timer for this interface. This specifies the maximum time in Milliseconds to wait before sending an acknowledgment for a sequenced PDU. A value of zero means there will be no delay in acknowledgement generation."

REFERENCE "ISO 8885 Table 3,

```
        Name: Reply delay timer"
DEFVAL { 0 }
 ::= { lapbAdmnEntry 10 }

lapbAdmnT3DisconnectTimer OBJECT-TYPE
    SYNTAX  PositiveInteger
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The T3 timer for this interface.  This
        specifies the time in Milliseconds to wait
        before considering the link disconnected.  A
        value of zero indicates the link will be
        considered disconnected upon completion of
        the frame exchange to disconnect the link."
    REFERENCE "ISO 7776 section 5.7.1.3"
    DEFVAL { 60000 }
    ::= { lapbAdmnEntry 11 }

lapbAdmnT4IdleTimer OBJECT-TYPE
    SYNTAX  PositiveInteger
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The T4 timer for this interface.  This
        specifies the maximum time in Milliseconds
        to allow without frames being exchanged on
        the data link.  A value of 2147483647
        indicates no idle timer is being kept."
    REFERENCE "ISO 7776 section 5.7.1.4"
    DEFVAL { 2147483647 }
    ::= { lapbAdmnEntry 12 }

lapbAdmnActionInitiate OBJECT-TYPE
    SYNTAX  INTEGER {
        sendSABM (1),
        sendDISC (2),
        sendDM (3),
        none (4),
        other (5)
    }
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "This identifies the action LAPB will take
        to initiate link set-up."
    DEFVAL { sendSABM }
    ::= { lapbAdmnEntry 13 }
```

## lapbAdmnActionRecvDM OBJECT-TYPE

```

SYNTAX  INTEGER {
    sendSABM (1),
    sendDISC (2),
    other (3)
}
ACCESS  read-write
STATUS  mandatory
DESCRIPTION
    "This identifies the action LAPB will take
    when it receives a DM response."
DEFVAL { sendSABM }
::= { lapbAdmnEntry 14 }

```

```

-- #####
--                                LAPB operating parameters.
-- #####

```

```

-- Support of the lapbOperTable is mandatory for all
-- agents of systems that implement LAPB.

```

## lapbOperTable OBJECT-TYPE

```

SYNTAX  SEQUENCE OF LapbOperEntry
ACCESS  not-accessible
STATUS  mandatory
DESCRIPTION
    "This table contains configuration
    information about interface parameters
    currently set in the interface.  Many of
    these objects have corresponding objects in
    the lapbAdmnTable."
::= { lapb 2 }

```

## lapbOperEntry OBJECT-TYPE

```

SYNTAX  LapbOperEntry
ACCESS  not-accessible
STATUS  mandatory
DESCRIPTION
    "Currently set parameter values for a
    specific LAPB."
INDEX { lapbOperIndex }
::= { lapbOperTable 1 }

```

```

LapbOperEntry ::= SEQUENCE {
    lapbOperIndex
        IfIndexType,
    lapbOperStationType

```

```

        INTEGER,
lapbOperControlField
        INTEGER,
lapbOperTransmitNlFrameSize
        PositiveInteger,
lapbOperReceiveNlFrameSize
        PositiveInteger,
lapbOperTransmitKWindowSize
        INTEGER,
lapbOperReceiveKWindowSize
        INTEGER,
lapbOperN2RxmitCount
        INTEGER,
lapbOperT1AckTimer
        PositiveInteger,
lapbOperT2AckDelayTimer
        PositiveInteger,
lapbOperT3DisconnectTimer
        PositiveInteger,
lapbOperT4IdleTimer
        PositiveInteger,
lapbOperPortId
        OBJECT IDENTIFIER,
lapbOperProtocolVersionId
        OBJECT IDENTIFIER
    }

lapbOperIndex      OBJECT-TYPE
    SYNTAX          IfIndexType
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The ifIndex value for the LAPB interface."
    ::= { lapbOperEntry 1 }

lapbOperStationType      OBJECT-TYPE
    SYNTAX INTEGER {
        dte (1),
        dce (2),
        dx (3)
    }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Identifies the current operating station
        type of this interface.  A value of dx (3)
        indicates XID negotiation has not yet taken
        place."

```

REFERENCE "ISO 7776 section 3.1"  
::= { lapbOperEntry 2 }

lapbOperControlField OBJECT-TYPE

SYNTAX INTEGER {  
    modulo8 (1),  
    modulo128 (2)  
}

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The current operating size of the sequence  
numbers used to number frames."

REFERENCE "ISO 7776 section 3.3"  
::= { lapbOperEntry 3 }

lapbOperTransmitNlFrameSize OBJECT-TYPE

SYNTAX PositiveInteger

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The current operating Nl frame size used  
for the maximum number of bits in a frame  
this DTE can transmit. This excludes flags  
and 0 bits inserted for transparency."

REFERENCE "ISO 7776 section 5.7.3"  
::= { lapbOperEntry 4 }

lapbOperReceiveNlFrameSize OBJECT-TYPE

SYNTAX PositiveInteger

ACCESS read-only

STATUS mandatory

-- See lapbOperTransmitNlFrameSize above

DESCRIPTION

"The current operating Nl frame size used  
for the maximum number of bits in a frame  
the DCE/remote DTE can transmit. This  
excludes flags and 0 bits inserted for  
transparency."

::= { lapbOperEntry 5 }

lapbOperTransmitKWindowSize OBJECT-TYPE

SYNTAX INTEGER (1..127)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The current PDU window size this Interface  
uses to transmit. This is the maximum

number of unacknowledged sequenced PDUs that may be outstanding from this DTE at any one time."

REFERENCE "ISO 7776 section 5.7.4"  
::= { lapbOperEntry 6 }

lapbOperReceiveKWindowSize OBJECT-TYPE

SYNTAX INTEGER (1..127)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The current receive PDU window size for this Interface. This is the maximum number of unacknowledged sequenced PDUs that may be outstanding from the DCE/remote DTE at any one time."

REFERENCE "ISO 7776 section 5.7.4"  
::= { lapbOperEntry 7 }

lapbOperN2RxmitCount OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The current N2 retry counter used for this interface. This specifies the number of times a PDU will be resent after the T1 timer expires without an acknowledgement for the PDU."

REFERENCE "ISO 7776 section 5.7.2"  
::= { lapbOperEntry 8 }

lapbOperT1AckTimer OBJECT-TYPE

SYNTAX PositiveInteger

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The current T1 timer for this interface. This specifies the maximum time in Milliseconds to wait for acknowledgment of a PDU."

REFERENCE "ISO 7776 section 5.7.1.1"  
::= { lapbOperEntry 9 }

lapbOperT2AckDelayTimer OBJECT-TYPE

SYNTAX PositiveInteger

ACCESS read-only

STATUS mandatory

## DESCRIPTION

"The current T2 timer for this interface. This specifies the maximum time in Milliseconds to wait before sending an acknowledgment for a sequenced PDU. A value of zero means there will be no delay in acknowledgement generation."

REFERENCE "ISO 7776 section 5.7.1.2"

::= { lapbOperEntry 10 }

## lapbOperT3DisconnectTimer OBJECT-TYPE

SYNTAX PositiveInteger

ACCESS read-only

STATUS mandatory

## DESCRIPTION

"The current T3 timer for this interface. This specifies the time in Milliseconds to wait before considering the link disconnected. A value of zero indicates the link will be considered disconnected upon completion of the frame exchange to disconnect the link."

REFERENCE "ISO 7776 section 5.7.1.3"

::= { lapbOperEntry 11 }

## lapbOperT4IdleTimer OBJECT-TYPE

SYNTAX PositiveInteger

ACCESS read-write

STATUS mandatory

## DESCRIPTION

"The current T4 timer for this interface. This specifies the maximum time in Milliseconds to allow without frames being exchanged on the data link. A value of 2147483647 indicates no idle timer is being kept."

REFERENCE "ISO 7776 section 5.7.1.4"

::= { lapbOperEntry 12 }

## lapbOperPortId OBJECT-TYPE

SYNTAX OBJECT IDENTIFIER

ACCESS read-only

STATUS mandatory

## DESCRIPTION

"This object identifies an instance of the index object in the first group of objects in the MIB specific to the physical device or interface used to send and receive



```

        frames.  If an agent does not support any
        such objects, it should return nullSpec
        OBJECT IDENTIFIER {0 0}."
 ::= { lapbOperEntry 13 }

lapbOperProtocolVersionId      OBJECT-TYPE
    SYNTAX  OBJECT IDENTIFIER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "This object identifies the version of the
        lapb protocol implemented by this
        interface."
 ::= { lapbOperEntry 14 }

-- #####
--                                LAPB      Flow Table
-- #####

-- Support of the lapbFlowTable is mandatory for all
-- agents of systems that implement LAPB.

lapbFlowTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF LapbFlowEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "This table defines the objects recorded by
        LAPB to provide information about the
        traffic flow through the interface."
 ::= { lapb 3 }

lapbFlowEntry OBJECT-TYPE
    SYNTAX  LapbFlowEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "The information regarding the effects of
        flow controls in LAPB."
    INDEX { lapbFlowIfIndex }
 ::= { lapbFlowTable 1 }

LapbFlowEntry ::= SEQUENCE {
    lapbFlowIfIndex
        IfIndexType,
    lapbFlowStateChanges
        Counter,

```

```

lapbFlowChangeReason
    INTEGER,
lapbFlowCurrentMode
    INTEGER,
lapbFlowBusyDefers
    Counter,
lapbFlowRejOutPkts
    Counter,
lapbFlowRejInPkts
    Counter,
lapbFlowTlTimeouts
    Counter,
lapbFlowFrmtSent
OCTET STRING,
lapbFlowFrmtReceived
    OCTET STRING,
lapbFlowXidReceived
    OCTET STRING
}

lapbFlowIfIndex OBJECT-TYPE
    SYNTAX  IfIndexType
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The ifIndex value for the LAPB Interface."
    ::= { lapbFlowEntry 1 }

lapbFlowStateChanges OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The number of LAPB State Changes, including
        resets."
    ::= { lapbFlowEntry 2 }

lapbFlowChangeReason OBJECT-TYPE
    SYNTAX  INTEGER {
        notStarted (1),      -- Initial state
        abmEntered (2),      -- SABM or UA
        abmeEntered (3),     -- SABME or UA
        abmReset (4),        -- SABM in ABM
        abmeReset (5),       -- SABME in ABME
        dmReceived (6),      -- DM Response
        dmSent (7),         -- DM sent
        discReceived (8),    -- DISC Response
        discSent (9),       -- DISC Sent
    }

```

```

        frmrReceived (10), -- FRMR Received
        frmrSent (11),    -- FRMR Sent
        n2Timeout (12),   -- N2 Timer Expired
        other (13)
    }
ACCESS   read-only
STATUS   mandatory
DESCRIPTION
    "The reason for the most recent incrementing
    of lapbFlowStateChanges. A DM or DISC frame
    generated to initiate link set-up does not
    alter this object. When the MIB-II object
    ifOperStatus does not have a value of
    testing, there exists a correlation between
    this object and ifOperStatus. IfOperStatus
    will have a value of up when this object
    contains: abmEntered, abmeEntered,
    abmReset, or abmeReset. IfOperStatus will
    have a value of down when this object has a
    value of notStarted, or dmReceived through
    n2Timeout. There is no correlation when
    this object has the value other."
 ::= { lapbFlowEntry 3 }

lapbFlowCurrentMode OBJECT-TYPE
    SYNTAX  INTEGER {
        disconnected (1),
            -- initial state or DISC received

        linkSetup (2),
            -- SABM sent

        frameReject (3),
            -- Invalid frame received and
            -- FRMR sent

        disconnectRequest (4),
            -- DISC sent

        informationTransfer (5),
            -- normal information transfer state
            -- SABM(E) sent and UA received, or
            -- SABM(E) received and UA sent

        rejFrameSent (6),
            -- invalid NS received and REJ sent

        waitingAcknowledgement (7),

```

```

        -- T1 expired and RR sent
stationBusy (8),
    -- RNR sent
remoteStationBusy (9),
    -- RNR received
bothStationsBusy (10),
    -- RNR received and RNR sent
waitingAckStationBusy (11),
    -- T1 expired, RNR sent
waitingAckRemoteBusy (12),
    -- T1 expired, RNR received
waitingAckBothBusy (13),
    -- T1 expired, RNR sent,
    -- and RNR received
rejFrameSentRemoteBusy (14),
    -- REJ sent and RNR received
xidFrameSent (15),
    -- XID frame sent
error (16),
    -- An error state other than
    -- a one defined above
other (17)
    -- A state not listed above
    }
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The current condition of the conversation."
::= { lapbFlowEntry 4 }

lapbFlowBusyDefers OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of times this device was unable
        to transmit a frame due to a perceived
        remote busy condition. Busy conditions can

```

```

        result from the receipt of an RNR from the
        remote device, the lack of valid sequence
        number space (window saturation), or other
        conditions."
 ::= { lapbFlowEntry 5 }

lapbFlowRejOutPkts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of REJ or SREJ frames sent by
        this station."
 ::= { lapbFlowEntry 6 }

lapbFlowRejInPkts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of REJ or SREJ frames received
        by this station."
 ::= { lapbFlowEntry 7 }

lapbFlowTlTimeouts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of times a re-transmission was
        effected by the Tl Timer expiring."
 ::= { lapbFlowEntry 8 }

lapbFlowFrmrSent OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE (0..7))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The Information Field of the FRMR most
        recently sent.  If no FRMR has been sent
        (the normal case) or the information isn't
        available, this will be an OCTET STRING of
        zero length."
    REFERENCE "ISO 7776 Section 4.3.9, tables 7 and 8"
 ::= { lapbFlowEntry 9 }

lapbFlowFrmrReceived OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE (0..7))
```

```

ACCESS    read-only
STATUS    mandatory
DESCRIPTION
    "The Information Field of the FRMR most
    recently received.  If no FRMR has been
    received (the normal case) or the
    information isn't available, this will be an
    OCTET STRING of zero length."
REFERENCE "ISO 7776 Section 4.3.9, tables 7 and 8"
::= { lapbFlowEntry 10 }

```

```

lapbFlowXidReceived OBJECT-TYPE
    SYNTAX  OCTET STRING (SIZE (0..8206))
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The Information Field of the XID frame most
        recently received.  If no XID frame has been
        received, this will be an OCTET STRING of
        zero length."
    REFERENCE "ISO 8885"
    ::= { lapbFlowEntry 11 }

```

```

-- #####
--                                LAPB      XID Table
-- #####

-- Support for the lapbXidTable is mandatory for all agents
-- of systems that have a LAPB implementation using XID
-- negotiation.  Agents of systems without XID negotiation
-- support should not implement this table.

```

```

lapbXidTable    OBJECT-TYPE
    SYNTAX  SEQUENCE OF LapbXidEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "This table defines values to use for XID
        negotiation that are not found in the
        lapbAdmnTable.  This table is optional for
        implementations that don't support XID and
        mandatory for implementations that do
        initiate XID negotiation."
    ::= { lapb 4 }

```

```

lapbXidEntry    OBJECT-TYPE
    SYNTAX  LapbXidEntry

```

```

ACCESS      not-accessible
STATUS      mandatory
DESCRIPTION
    "Xid negotiation parameter values for a
    specific LAPB."
INDEX { lapbXidIndex }
::= { lapbXidTable 1 }

LapbXidEntry ::= SEQUENCE {
    lapbXidIndex
        IfIndexType,
    lapbXidAdrIdentifier
        OCTET STRING,
    lapbXidAdrAddress
        OCTET STRING,
    lapbXidParameterUniqueIdentifier
        OCTET STRING,
    lapbXidGroupAddress
        OCTET STRING,
    lapbXidPortNumber
        OCTET STRING,
    lapbXidUserDataSubfield
        OCTET STRING
}

lapbXidIndex      OBJECT-TYPE
SYNTAX      IfIndexType
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
    "The ifIndex value for the LAPB interface."
::= { lapbXidEntry 1 }

lapbXidAdrIdentifier OBJECT-TYPE
SYNTAX      OCTET STRING (SIZE (0..255))
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
    "The value of the Address Resolution
    Identifier. A zero length string indicates
    no Identifier value has been assigned."
REFERENCE "ISO 8885 Table 2, Name: Identifier"
DEFVAL { ''h }
::= { lapbXidEntry 2 }

lapbXidAdrAddress OBJECT-TYPE
SYNTAX      OCTET STRING (SIZE (0..255))

```

ACCESS read-write  
STATUS mandatory  
DESCRIPTION  
    "The value of the Address Resolution  
    Address. A zero length string indicates no  
    Address value has been assigned."  
REFERENCE "ISO 8885 Table 2, Name: Address"  
DEFVAL { ''h }  
::= { lapbXidEntry 3 }

lapbXidParameterUniqueIdentifier OBJECT-TYPE  
SYNTAX OCTET STRING (SIZE (0..255))  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION  
    "The value of the parameter unique  
    Identifier. A zero length string indicates  
    no Unique identifier value has been  
    assigned."  
REFERENCE "ISO 8885 Table 3, Name: Identifier"  
DEFVAL { ''h }  
::= { lapbXidEntry 4 }

lapbXidGroupAddress OBJECT-TYPE  
SYNTAX OCTET STRING (SIZE (0..255))  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION  
    "The value of the parameter Group address.  
    A zero length string indicates no Group  
    address value has been assigned."  
REFERENCE "ISO 8885 Table 3, Name: Group address"  
DEFVAL { ''h }  
::= { lapbXidEntry 5 }

lapbXidPortNumber OBJECT-TYPE  
SYNTAX OCTET STRING (SIZE (0..255))  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION  
    "The port number assigned for this link. A  
    zero length string indicates no local port  
    number identifier has been assigned."  
REFERENCE "ISO 8885 Table 3, Name: Port number"  
DEFVAL { ''h }  
::= { lapbXidEntry 6 }

lapbXidUserDataSubfield OBJECT-TYPE



```

SYNTAX  OCTET STRING (SIZE (0..8206))
ACCESS  read-write
STATUS  mandatory
DESCRIPTION
    "A user data subfield, if any, to be
    transmitted in an XID frame.  A zero length
    frame indicates no user data subfield has
    been assigned.  The octet string should
    include both the User data identifier and
    User data field as shown in Figures 1 and
    4."
REFERENCE "ISO 8885 section 4.3"
DEFVAL { 'h' }
::= { lapbXidEntry 7 }

-- #####
--                               LAPB protocol versions
-- #####

lapbProtocolVersion OBJECT IDENTIFIER
::= { lapb 5 }

lapbProtocolIso7776v1986 OBJECT IDENTIFIER
::= { lapbProtocolVersion 1 }

lapbProtocolCcittV1980 OBJECT IDENTIFIER
::= { lapbProtocolVersion 2 }

lapbProtocolCcittV1984 OBJECT IDENTIFIER
::= { lapbProtocolVersion 3 }

-- The following describes some of the MIB-II interface
-- objects and their relationship with the objects in this
-- MIB extension.

-- ifDescr:  describes the interface.  It should include
--            identification information for the physical line and a
--            description of the network.  For connections to PDNs,
--            it should name the PDN.

-- ifMtu: the maximum number of octets an upper layer can
-- pass to this interface as a single frame.

-- ifSpeed:

```

```
-- ifAdminStatus:

-- ifOperStatus:

-- ifLastChange: the last time the state of the interface
-- changed. A reset is considered an instantaneous change to
-- the ndm state and back to abm or abme. This will be the
-- last time that lapbFlowChangeReason and lapbFlowChanges
-- changed.

-- ifInOctets: contains the number of octets
-- received from the peer LAPB including FCS.

-- ifInUcastPkts: contains the number of I-frames delivered
-- by this interface to a higher layer interface.

-- ifInDiscards: contains the number of received
-- frames discarded because of internal conditions
-- (such as lack of buffering).

-- ifInErrors: contains the number of Invalid frames received.
-- This does not have any relationship with the number REJ,
-- or RNR frames sent or received.

-- ifInUnknownProtos: contains the number of frames
-- that were correct but were dropped because they
-- were inappropriate for the current state. This
-- includes an invalid Poll bit, an unknown address,
-- or other condition such as an RNR when connection
-- not established. This also includes the number of
-- DISC or other frames that were ignored because the
-- link was not established and this interface was not
-- configured to perform link setup on that type frame.

-- ifOutOctets: number of octets sent to peer including
-- FCS octets.

-- ifOutUcastPkts: number of I-frames received from
-- a higher layer for transmission to peer.

-- ifOutDiscards: number of frames to be sent that were
-- dropped due to internal conditions such as buffering etc.

-- ifOutErrors: number of transmissions that failed
-- due to errors or were considered invalid by the receiver.
-- This does not have any relationship with the number REJ,
-- or RNR frames sent or received.
```

```
-- ifOutQLen: number of frames waiting to be transmitted.

-- This MIB does not provide any support for:
--     Multilink procedure (MLP) in ISO 7776 section 6
--     LLC Pbit timer
--     LLC REJ timer
--     LLC Busy State Timer 7.8.1.4

-- #####

END
```

## 5. Appendix: Revision History

July 30, 1992

The July revision of this document (Editor's Internal Reference 2.10) incorporated the comments of the SNMP directorate.

The ifIndexType textual convention was added and used as the type for all index objects.

The enumeration xidDetection of the lapbAdmnStationType was changed to dxex to be consistent with other similar enumerations.

Conformance statements were added at before every table as ASN.1 comments.

June 12, 1992

The June 12, 1992 revision of this document (Editor's Internal Reference 2.9) incorporated some clarifications and updated the status.

The range on PositiveInteger was changed to start at 0 rather than 1.

The syntax of lapbXidIndex was changed to PositiveInteger.

A value of dxex was added to lapbOperStationType.

The range of lapbAdmnN2RxmitCount was change to (0..65535).

The definition of ifInOctets, ifInUcastPkts, ifInErrors, ifInUnknownProtos, ifOutOctets, and ifoutUcastPkts was clarified.

May 18, 1992

The May 18, 1992 revision of this document (Editor's Internal Reference 2.8) incorporated the following changes:

The states of lapbFlowCurrentMode were redefined.

The default value for lapbAdmnControlField was changed from module8 to modulo8.

April 8, 1992

The April 8, 1992 revision of this document (Editor's Internal Reference 2.4) incorporated the following changes:

All reference comments in the MIB were moved to the REFERENCE field of the OBJECT-TYPE macro.

A type of PositiveInteger was introduced and used for common integer values including all timers. This effectively made the maximum value for timers 2147483646 milliseconds. The type of the frame size was changed to positiveInteger.

The reference to ISO 7776 has been broadened to say the MIB descriptions use the terminology of ISO 7776.

A comment was added to the overview section discussing creation and deletion of tables.

The objects in the lapbParmTable and lapbDefTable were redistributed to create a lapbOperTable, a lapbAdmnTable, and a lapbXidTable. The lapbParmTable and lapbDefTable were deleted. Objects were included in the Admn table for t3 and t4.

An object identifier was added to identify the protocol version.

A DEFVAL clause was added for all writable objects.

Some more overview text was included.

February 1992

The February 1992 revision of this document (Editor's Internal Reference 1.17) incorporated the following changes:

The name was changed from HDLC to LAPB. This change was made because other flavors of HDLC such as LAPD, SDLC, and raw HDLC framing, are different enough that this MIB will not adequately

manage them.

The Historical Perspective section at the beginning of the document has been replaced with a more concise Network Management Framework section.

The name lapbParmKWindowSize was changed to lapbParmTransmitKWindowSize and the object lapbParmReceiveKWindowSize was added. This change was made because section 5.7.4 of ISO 7776 and Table 3 of ISO 8885 have provisions for different values for the transmit and receive window size.

The name lapbParmNlFrameSize was changed to lapbParmTransmitNlFrameSize and the object lapbParmReceiveNlFrameSize was added. This change was made because section 5.7.3 of ISO 7776 and Table 3 of ISO 8886 have provisions for different values for the transmit and receive maximum frame size.

The object lapbParmPortIndex was deleted and the description of lapbParmPortId was changed. The object lapbParmPortId now identifies an instance of the index object for the MIB of the physical device or interface below LAPB.

The units for the timers were changed to Milliseconds to be consistent with ISO 8885; see table 3.

The objects lapbParamT2AckDelayTimer and lapbParamT3DisconnectTimer both allow values of 0 to indicate the timer is not being used.

The object lapbParamT4IdleTimer has a value to indicate timer not in use.

The object lapbFlowXidReceived was added to the flow table.

The lapbDefTable was added.

Ranges and sizes were added for all INTEGERS and OCTET STRINGS that didn't have them.

October 1991

The October 1991 revision of this document basically changed the name from LAPB to HDLC to make the objects more appropriate for a broader range of uses. A number of minor changes were made to bring the objects in line with established conventions. These changes are as follows.

The enumerated values of `hdlcParmStationType` were renumbered from 0 and 1 to 1 and 2.

The object `hdlcFlowBusyDefer` was renamed `hdlcFlowBusyDefers`.

The object `hdlcFlowRejSent` was rename `hdlcFlowRejOutPkts`.

The object `hdlcFlowRejReceived` was renamed `hdlcFlowRejInPkts`.

June 1991

The June revision of this document incorporated much of the E-mail discussion of the first draft. In particular it replaced the `lapbStatTable` (and all contents) with the `lapbFlowTable`.

April 1991

The April 24 version of this document was the first release. At that time this document was basically a bunch of objects synthesized from various vendor MIBs and a quick reading of ISO 7776 [10]. On first reading it appeared to instrument too many LAPB normal functions and too few exceptional conditions. The `lapbStatTable` was too long and needed to be redone.

## 6. Acknowledgements

This document was produced by the x25mib working group:

Fred Baker, ACC  
Art Berggreen, ACC  
Frank Bieser  
Gary Bjerke, Tandem  
Bill Bowman, HP  
Christopher Bucci, Datability  
Charles Carvalho, ACC  
Jeff Case, Snmp Research  
Angela Chen, HP  
Carson Cheung, BNR  
Tom Daniel, Spider Systems  
Chuck Davin, MIT  
Billy Durham, Honeywell  
Richard Fox, Synoptics  
Doug Geller, Data General  
Herve Goguely, LIR Corp  
Andy Goldthorpe, british-telecom  
Walter D. Guilarte  
David Gurevich  
Steve Huston, Process Software Corporation

Jon Infante, ICL  
Frank Kastenholz, Clearpoint  
Zbigniew Kielczewski, Eicon  
Cheryl Krupezak, Georgia Tech  
Mats Lindstrom, Diab Data AB  
Andrew Malis, BBN  
Evan McGinnis, 3Com  
Gary (G.P.)Mussar, BNR  
Chandy Nilakantan, 3Com  
Randy Pafford, Data General  
Ragnar Paulson, The Software Group Limited  
Dave Perkins, Synoptics  
Walter Pinkarschewsky, DEC  
Karen Quidley, Data General  
Chris Ranch, Novell  
Paul S. Rarey, DHL Systems Inc.  
Jim Roche, Newbridge Research  
Philippe Roger, LIR Corp.  
Timon Sloane  
Mike Shand, DEC  
Brad Steina, Microcom  
Bob Stewart, Xyplex  
Tom Sullivan, Data General  
Rodney Thayer, Sable Technology Corporation  
Mark Therieau, Microcom  
Jane Thorn, Data General  
Dean Throop, Data General  
Maurice Turcotte, Racal Datacom  
Mike Zendels, Data General

In addition, the comments of the following individuals are also acknowledged:

Keith McCloghrie

## 7. References

- [1] Rose M., and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based internets", STD 16, RFC 1155, Performance Systems International, Hughes LAN Systems, May 1990.
- [2] 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.
- [3] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple

Network Management Protocol", STD 15, RFC 1157, SNMP Research, Performance Systems International, Performance Systems International, MIT Laboratory for Computer Science, May 1990.

- [4] Rose, M., and K. McCloghrie, Editors, "Concise MIB Definitions", STD 16, RFC 1212, Performance Systems International, Hughes LAN Systems, March 1991.
- [5] Rose M., Editor, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", STD 17, RFC 1213, Performance Systems International, March 1991.
- [6] Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1), International Organization for Standardization, International Standard 8824, December 1987.
- [7] 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.
- [8] Stewart, B., Editor, "Definitions of Managed Objects for RS-232-like Hardware Devices", RFC 1317, Xyplex, Inc., April 1992.
- [9] Throop, D., Editor, "SNMP MIB extension for the Packet Layer of X.25", RFC 1382, Data General Corporation, November 1992.
- [10] "Information processing systems - Data communication - High-level data link control procedure - Description of the X.25 LAPB-compatible DTE data link procedures", International Organization for Standardization, International Standard 7776, December 1986.
- [11] "Information technology - Telecommunications and information exchange between systems - High-level data link control (HDLC) procedures - General purpose XID frame information field contents and format", International Organization for Standardization, International Standard 8885.



## 8. Security Considerations

Security issues are not discussed in this memo.

## 9. Authors' Addresses

Dean D. Throop  
Data General Corporation  
62 Alexander Dr.  
Research Triangle Park, NC 27709

Phone: (919)248-8421  
EMail: throop@dg-rtp.dg.com

Fred Baker  
Advanced Computer Communications  
315 Bollay Drive  
Santa Barbara, CA 93101

Phone: (805) 685-4455  
EMail: fbaker@acc.com

While the working group has completed discussion of this document, comments are still welcome. Please send comments to the x25mib working group at: x25mib@dg-rtp.dg.com