

Network Working Group
Request for Comments: 1315

C. Brown
Wellfleet Communications, Inc.
F. Baker
Advanced Computer Communications
C. Carvalho
Advanced Computer Communications
April 1992

Management Information Base for Frame Relay DTEs

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 Frame Relay.

Table of Contents

1. The Network Management Framework	2
2. Objects	2
2.1 Format of Definitions	3
3. Overview	3
3.1 Frame Relay Operational Model	3
3.2 Textual Conventions	3
3.3 Structure of MIB	3
4. Definitions	4
4.1 Data Link Connection Management Interface	4
4.2 Circuit Table	9
4.3 Error Table	14
5. Acknowledgements	17
6. References	17
7. Security Considerations.....	18
8. Authors' Addresses.....	19

1. 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.

2. 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.

2.1. Format of Definitions

Section 4 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 [9,10].

3. Overview

3.1. Frame Relay Operational Model

For the purposes of understanding this document, Frame Relay is viewed as a multi-access media, not as a group of point-to-point connections. This model proposes that Frame Relay is a single interface to the network (physical connection) with many destinations or neighbors (virtual connections). This view enables a network manager the ability to group all virtual connections with their corresponding physical connection thereby allowing simpler diagnostics and trouble shooting.

3.2. Textual Conventions

Several new data types are introduced as a textual convention 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 the 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.

The new data types are Index and DLCI. Index refers to the range 1..ifNumber, and is used to establish the correspondence between ifEntries and Frame Relay Interfaces. DLCI refers to the range 0..DLCINumber, and is used to refer to the valid Data Link Connection Indices. DLCINumber is, by definition, the largest possible DLCI value possible under the configured Q.922 Address Format.

3.3. Structure of MIB

The MIB is composed of three groups, one defining the Data Link Connection Management Interface (DLCMI), one describing the Circuits, and a third describing errors.

During normal operation, Frame Relay virtual circuits will be added, deleted and change availability. The occurrence of such changes is of interest to the network manager and therefore, one trap is defined, intended to be corollary to the SNMP "Link Up" and "Link Down" traps.

4. Definitions

```
RFC1315-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    OBJECT-TYPE
        FROM RFC-1212
    transmission
        FROM RFC1213-MIB
    TimeTicks
        FROM RFC-1155
    TRAP-TYPE
        FROM RFC-1215;
```

```
--   Frame Relay DTE MIB
```

```
frame-relay      OBJECT IDENTIFIER ::= { transmission 32 }
```

```
--
```

```
--       the range of ifIndex
```

```
--
```

```
Index ::= INTEGER      -- 1..ifNumber
```

```
--
```

```
--       the range of a Data Link Connection Identifier
```

```
--
```

```
DLCI ::= INTEGER      -- 0..DLCINumber
```

```
--   Data Link Connection Management Interface
```

```
--       The variables that configure the DLC Management Interface.
```

```
frDlcmiTable OBJECT-TYPE
```

```
    SYNTAX      SEQUENCE OF FrDlcmiEntry
```

```
    ACCESS      not-accessible
```

```
    STATUS      mandatory
```

```
    DESCRIPTION
```

```
        "The Parameters for the Data Link Connection Management
         Interface for the frame relay service on this
         interface."
```

```
    REFERENCE
```

"Draft American National Standard T1.617-1991, Annex D"
 ::= { frame-relay 1 }

frDlcmiEntry OBJECT-TYPE
 SYNTAX FrDlcmiEntry
 ACCESS not-accessible
 STATUS mandatory
 DESCRIPTION
 "The Parameters for a particular Data Link Con-
 nection Management Interface."
 INDEX { frDlcmiIfIndex }
 ::= { frDlcmiTable 1 }

FrDlcmiEntry ::=
 SEQUENCE {
 frDlcmiIfIndex
 Index,
 frDlcmiState
 INTEGER,
 frDlcmiAddress
 INTEGER,
 frDlcmiAddressLen
 INTEGER,
 frDlcmiPollingInterval
 INTEGER,
 frDlcmiFullEnquiryInterval
 INTEGER,
 frDlcmiErrorThreshold
 INTEGER,
 frDlcmiMonitoredEvents
 INTEGER,
 frDlcmiMaxSupportedVCs
 INTEGER,
 frDlcmiMulticast
 INTEGER
 }

frDlcmiIfIndex OBJECT-TYPE
 SYNTAX Index
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "The ifIndex value of the corresponding ifEn-
 try."
 ::= { frDlcmiEntry 1 }

frDlcmiState OBJECT-TYPE

SYNTAX INTEGER {

noLmiConfigured (1),

lmiRev1 (2),

ansiT1-617-D (3), -- ANSI T1.617 Annex D

ansiT1-617-B (4) -- ANSI T1.617 Annex B

}

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This variable states which Data Link Connection Management scheme is active (and by implication, what DLCI it uses) on the Frame Relay interface."

REFERENCE

"Draft American National Standard T1.617-1991"

::= { frDlcmiEntry 2 }

frDlcmiAddress OBJECT-TYPE

SYNTAX INTEGER {

q921 (1), -- 13 bit DLCI

q922March90 (2), -- 11 bit DLCI

q922November90 (3), -- 10 bit DLCI

q922 (4) -- Final Standard

}

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This variable states which address format is in use on the Frame Relay interface."

::= { frDlcmiEntry 3 }

frDlcmiAddressLen OBJECT-TYPE

SYNTAX INTEGER {

two-octets (2),

three-octets (3),

four-octets (4)

}

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This variable states which address length in octets. In the case of Q922 format, the length indicates the entire length of the address including the control portion."

::= { frDlcmiEntry 4 }

frDlcmiPollingInterval OBJECT-TYPE

SYNTAX INTEGER (5..30)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This is the number of seconds between successive status enquiry messages."

REFERENCE

"Draft American National Standard T1.617-1991, Section D.7 Timer T391."

DEFVAL { 10 }

::= { frDlcmiEntry 5 }

frDlcmiFullEnquiryInterval OBJECT-TYPE

SYNTAX INTEGER (1..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Number of status enquiry intervals that pass before issuance of a full status enquiry message."

REFERENCE

"Draft American National Standard T1.617-1991, Section D.7 Counter N391."

DEFVAL { 6 }

::= { frDlcmiEntry 6 }

frDlcmiErrorThreshold OBJECT-TYPE

SYNTAX INTEGER (1..10)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This is the maximum number of unanswered Status Enquiries the equipment shall accept before declaring the interface down."

REFERENCE

"Draft American National Standard T1.617-1991, Section D.5.1 Counter N392."

DEFVAL { 3 }

::= { frDlcmiEntry 7 }

frDlcmiMonitoredEvents OBJECT-TYPE

SYNTAX INTEGER (1..10)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This is the number of status polling intervals over which the error threshold is counted. For example, if within 'MonitoredEvents' number of events the station receives 'ErrorThreshold' number of errors, the interface is marked as down."

REFERENCE

"Draft American National Standard T1.617-1991, Section D.5.2 Counter N393."

DEFVAL { 4 }

::= { frDlcmiEntry 8 }

frDlcmiMaxSupportedVCs OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

DESCRIPTION

"The maximum number of Virtual Circuits allowed for this interface. Usually dictated by the Frame Relay network."

In response to a SET, if a value less than zero or higher than the agent's maximal capability is configured, the agent should respond bad-Value"

::= { frDlcmiEntry 9 }

frDlcmiMulticast OBJECT-TYPE

SYNTAX INTEGER {
 nonBroadcast (1),
 broadcast (2)
}

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This indicates whether the Frame Relay interface is using a multicast service."

::= { frDlcmiEntry 10 }


```
-- A Frame Relay service is a multiplexing service.  Data
-- Link Connection Identifiers enumerate virtual circuits
-- (permanent or dynamic) which are layered onto the underlying
-- circuit, represented by ifEntry.  Therefore, each of the entries
-- in the Standard MIB's Interface Table with an IfType of
-- Frame Relay represents a Q.922 interface.  Zero or more
-- virtual circuits are layered onto this interface and provide
-- interconnection with various remote destinations.
-- Each such virtual circuit is represented by an entry in the
-- circuit table.
```

```
--  Circuit Table
```

```
-- The table describing the use of the DLCIs attached to
-- each Frame Relay Interface.
```

```
frCircuitTable OBJECT-TYPE
```

```
    SYNTAX      SEQUENCE OF FrCircuitEntry
```

```
    ACCESS      not-accessible
```

```
    STATUS      mandatory
```

```
    DESCRIPTION
```

```
        "A table containing information about specific Data
        Link Connection Identifiers and corresponding virtual
        circuits."
```

```
    ::= { frame-relay 2 }
```

```
frCircuitEntry OBJECT-TYPE
```

```
    SYNTAX      FrCircuitEntry
```

```
    ACCESS      not-accessible
```

```
    STATUS      mandatory
```

```
    DESCRIPTION
```

```
        "The information regarding a single Data Link
        Connection Identifier."
```

```
    INDEX { frCircuitIfIndex, frCircuitDlci }
```

```
    ::= { frCircuitTable 1 }
```

```
FrCircuitEntry ::=
```

```
    SEQUENCE {
```

```
        frCircuitIfIndex
        Index,
```

```
        frCircuitDlci
```

```
        DLCI,
```

```
        frCircuitState
```

```
        INTEGER,
```

```
        frCircuitReceivedFECNs
```

```
        Counter,
```

```
        frCircuitReceivedBECNs
```

```

        Counter,
    frCircuitSentFrames
        Counter,
    frCircuitSentOctets
        Counter,
    frCircuitReceivedFrames
        Counter,
    frCircuitReceivedOctets
        Counter,
    frCircuitCreationTime
        TimeTicks,
    frCircuitLastTimeChange
        TimeTicks,
    frCircuitCommittedBurst
        INTEGER,
    frCircuitExcessBurst
        INTEGER,
    frCircuitThroughput
        INTEGER
}

```

frCircuitIfIndex OBJECT-TYPE

SYNTAX Index

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The ifIndex Value of the ifEntry this virtual circuit is layered onto."

::= { frCircuitEntry 1 }

frCircuitDlci OBJECT-TYPE

SYNTAX DLCI

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The Data Link Connection Identifier for this virtual circuit."

REFERENCE

"Draft American National Standard T1.618-1991, Section 3.3.6"

::= { frCircuitEntry 2 }

```
frCircuitState OBJECT-TYPE
    SYNTAX      INTEGER {
                    invalid (1),
                    active (2),
                    inactive (3)
                }
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION
        "Indicates whether the particular virtual circuit is operational. In the absence of a Data Link Connection Management Interface, virtual circuit entries (rows) may be created by setting virtual circuit state to 'active', or deleted by changing Circuit state to 'invalid'. Whether or not the row actually disappears is left to the implementation, so this object may actually read as 'invalid' for some arbitrary length of time. It is also legal to set the state of a virtual circuit to 'inactive' to temporarily disable a given circuit."
    DEFVAL { active }
    ::= { frCircuitEntry 3 }
```

```
frCircuitReceivedFECNs OBJECT-TYPE
    SYNTAX      Counter
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "Number of frames received from the network indicating forward congestion since the virtual circuit was created."
    REFERENCE
        "Draft American National Standard T1.618-1991, Section 3.3.3"
    ::= { frCircuitEntry 4 }
```

```
frCircuitReceivedBECNs OBJECT-TYPE
    SYNTAX      Counter
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION
        "Number of frames received from the network indicating backward congestion since the virtual circuit was created."
```

REFERENCE

"Draft American National Standard T1.618-1991,
Section 3.3.4"
::= { frCircuitEntry 5 }

frCircuitSentFrames OBJECT-TYPE

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The number of frames sent from this virtual
circuit since it was created."

::= { frCircuitEntry 6 }

frCircuitSentOctets OBJECT-TYPE

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The number of octets sent from this virtual
circuit since it was created."

::= { frCircuitEntry 7 }

frCircuitReceivedFrames OBJECT-TYPE

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Number of frames received over this virtual
circuit since it was created."

::= { frCircuitEntry 8 }

frCircuitReceivedOctets OBJECT-TYPE

SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Number of octets received over this virtual
circuit since it was created."

::= { frCircuitEntry 9 }

frCircuitCreationTime OBJECT-TYPE

SYNTAX TimeTicks
ACCESS read-only
STATUS mandatory

DESCRIPTION

"The value of sysUpTime when the virtual circuit was created, whether by the Data Link Connection Management Interface or by a SetRequest."

::= { frCircuitEntry 10 }

frCircuitLastTimeChange OBJECT-TYPE

SYNTAX TimeTicks
ACCESS read-only
STATUS mandatory

DESCRIPTION

"The value of sysUpTime when last there was a change in the virtual circuit state"

::= { frCircuitEntry 11 }

frCircuitCommittedBurst OBJECT-TYPE

SYNTAX INTEGER
ACCESS read-write
STATUS mandatory

DESCRIPTION

"This variable indicates the maximum amount of data, in bits, that the network agrees to transfer under normal conditions, during the measurement interval."

REFERENCE

"Draft American National Standard T1.617-1991, Section 6.5.19"

DEFVAL { 0 } -- the default indicates no commitment

::= { frCircuitEntry 12 }

frCircuitExcessBurst OBJECT-TYPE

SYNTAX INTEGER
ACCESS read-write
STATUS mandatory

DESCRIPTION

"This variable indicates the maximum amount of uncommitted data bits that the network will at-

tempt to deliver over the measurement interval.

By default, if not configured when creating the entry, the Excess Information Burst Size is set to the value of ifSpeed."

REFERENCE

"Draft American National Standard T1.617-1991, Section 6.5.19"

::= { frCircuitEntry 13 }

frCircuitThroughput OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Throughput is the average number of 'Frame Relay Information Field' bits transferred per second across a user network interface in one direction, measured over the measurement interval.

If the configured committed burst rate and throughput are both non-zero, the measurement interval

$T = \text{frCircuitCommittedBurst} / \text{frCircuitThroughput}$.

If the configured committed burst rate and throughput are both zero, the measurement interval

$T = \text{frCircuitExcessBurst} / \text{ifSpeed}$."

REFERENCE

"Draft American National Standard T1.617-1991, Section 6.5.19"

DEFVAL {0} -- the default value of Throughput is
-- "no commitment".

::= { frCircuitEntry 14 }

-- Error Table

-- The table describing errors encountered on each Frame
-- Relay Interface.

frErrTable OBJECT-TYPE

SYNTAX SEQUENCE OF FrErrEntry

ACCESS not-accessible

STATUS mandatory
DESCRIPTION
"A table containing information about Errors on the
Frame Relay interface."
::= { frame-relay 3 }

frErrEntry OBJECT-TYPE
SYNTAX FrErrEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"The error information for a single frame relay
interface."
INDEX { frErrIfIndex }
::= { frErrTable 1 }

FrErrEntry ::=
SEQUENCE {
frErrIfIndex
Index,
frErrType
INTEGER,
frErrData
OCTET STRING,
frErrTime
TimeTicks
}

frErrIfIndex OBJECT-TYPE
SYNTAX Index
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The ifIndex Value of the corresponding ifEn-
try."
::= { frErrEntry 1 }

frErrType OBJECT-TYPE
SYNTAX INTEGER {
unknownError(1),
receiveShort(2),
receiveLong(3),

```

        illegalDLCI(4),
        unknownDLCI(5),
        dlcmiProtoErr(6),
        dlcmiUnknownIE(7),
        dlcmiSequenceErr(8),
        dlcmiUnknownRpt(9),
        noErrorSinceReset(10)
    }
ACCESS    read-only
STATUS    mandatory
DESCRIPTION
    "The type of error that was last seen on this
    interface."
::= { frErrEntry 2 }

```

```

frErrData OBJECT-TYPE
    SYNTAX    OCTET STRING
    ACCESS    read-only
    STATUS    mandatory
    DESCRIPTION
        "An octet string containing as much of the er-
        ror packet as possible. As a minimum, it must
        contain the Q.922 Address or as much as was
        delivered. It is desirable to include all in-
        formation up to the PDU."
    ::= { frErrEntry 3 }

```

```

frErrTime OBJECT-TYPE
    SYNTAX    TimeTicks
    ACCESS    read-only
    STATUS    mandatory
    DESCRIPTION
        "The value of sysUpTime at which the error was
        detected."
    ::= { frErrEntry 4 }

```

```
-- Frame Relay Globals
```

```
frame-relay-globals OBJECT IDENTIFIER ::= { frame-relay 4 }
```

```

frTrapState OBJECT-TYPE
    SYNTAX    INTEGER { enabled(1), disabled(2) }
    ACCESS    read-write

```



```
STATUS    mandatory
DESCRIPTION
    "This variable indicates whether the system
    produces the frDLCIStatusChange trap."
DEFVAL { disabled }
 ::= { frame-relay-globals 1 }
```

-- Data Link Connection Management Interface Related Traps

```
frDLCIStatusChange TRAP-TYPE
ENTERPRISE frame-relay
VARIABLES { frCircuitIfIndex, frCircuitDlci, frCircuitState }
DESCRIPTION
    "This trap indicates that the indicated Virtual
    Circuit has changed state. It has either been
    created or invalidated, or has toggled between
    the active and inactive states."
 ::= 1
```

END

5. Acknowledgements

This document was produced by the IP Over Large Public Data Networks (IPLPDN) Working Group.

The following people provided additional comments and suggestions: Art Berggreen of Advanced Computer Communications, and Jim Philippou of Xyplex Communications.

6. 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] Rose, M., and K. McCloghrie, Editors, "Concise MIB Definitions", RFC 1212, Performance Systems International, Hughes LAN Systems, March 1991.
- [10] Rose, M., Editor, "A Convention for Defining Traps for use with the SNMP", RFC 1215, Performance Systems International, March 1991.

7. Security Considerations

Security issues are not discussed in this memo.

8. Authors' Addresses

Caralyn Brown
Wellfleet Communications, Inc.
15 Crosby Drive
Bedford, Massachusetts 01730

Phone: (617) 275-2400
EMail: cbrown@wellfleet.com

Fred Baker
Advanced Computer Communications
315 Bollay Drive
Sannta Barbara, California 93117

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

Charles Carvalho
Advanced Computer Communications
315 Bollay Drive
Sannta Barbara, California 93117

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