

CLNS MIB  
for use with  
Connectionless Network Protocol (ISO 8473)  
and  
End System to Intermediate System (ISO 9542)

Status of this Memo

This memo defines an experimental portion of the Management Information Base (MIB) for use with network management protocols in TCP/IP-based internets. This is an Experimental Protocol for the Internet community. Discussion and suggestions for improvement are requested. 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. The Network Management Framework.....	1
2. Objects .....	2
2.1 Format of Definitions .....	2
3. Overview .....	2
3.1 Textual Conventions .....	3
3.2 Changes from RFC 1162 .....	3
4. Definitions .....	4
4.1 Textual Conventions .....	4
4.2 Groups in the CLNS MIB .....	4
4.3 The CLNP Group .....	4
4.4 The CLNP Error Group .....	21
4.5 The ES-IS Group .....	30
5. References .....	31
6. Security Considerations .....	32
7. Author's Address.....	32

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 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].

## 3. Overview

The objects defined in this MIB module are to be used when the ISO Connectionless-mode Network Protocol [10] and End System to

Intermediate System [11] protocols are present. No assumptions are made as to what underlying protocol is being used to carry the SNMP.

This memo uses the string encoding of [12] to textually describe OSI addresses.

### 3.1. Textual Conventions

A new data type is introduced as a textual convention in this MIB document. This 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 this textual convention has no effect on either the syntax nor the semantics of any managed objects. The use of this is merely an artifact of the explanatory method used. Objects defined in terms of this 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 this textual convention 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 ASN.1 type ClnpAddress is used to denote an OSI address. This consists of a string of octets. The first octet of the string contains a binary value in the range of 0..20, and indicates the the length in octets of the NSAP. Following the first octet, is the NSAP, expressed in concrete binary notation, starting with the most significant octet. A zero- length NSAP is used as a "special" address meaning "the default NSAP" (analogous to the IP address of 0.0.0.0). Such an NSAP is encoded as a single octet, containing the value 0. All other NSAPs are encoded in at least 4 octets.

### 3.2. Changes from RFC 1162

Features of this MIB include:

- (1) The managed objects in this document have been defined using the conventions defined in the Internet-standard SMI, as amended by the extensions specified in [9]. It must be emphasized that definitions made using these extensions are semantically identically to those in RFC 1162.
- (2) The PhysAddress textual convention from MIB-II has been introduced to represent media addresses.
- (3) The clnpRoutingDiscards, clnpRouteMetric5, and clnpRouteInfo objects have been defined.

- (4) The ClnpAddress type was mistakenly given a tag in RFC 1162. This error has been corrected.

#### 4. Definitions

```
CLNS-MIB DEFINITIONS ::= BEGIN

IMPORTS
    experimental, Counter
        FROM RFC1155-SMI
    PhysAddress
        FROM RFC-1213
    OBJECT-TYPE
        FROM RFC-1212;

-- This MIB module uses the extended OBJECT-TYPE macro as
-- defined in [9]

-- the CLNS MIB module
clns    OBJECT IDENTIFIER ::= { experimental 1 }

-- textual conventions
ClnpAddress ::=
    OCTET STRING (SIZE (1..21))
-- This data type is used to model NSAP addresses.

-- groups in the CLNS MIB
clnp    OBJECT IDENTIFIER ::= { clns 1 }
error   OBJECT IDENTIFIER ::= { clns 2 }
echo    OBJECT IDENTIFIER ::= { clns 3 }
es-is   OBJECT IDENTIFIER ::= { clns 4 }

-- the CLNP group
-- Implementation of this group is recommended for all
-- systems which implement the CLNP.
```

```

clnpForwarding OBJECT-TYPE
    SYNTAX  INTEGER {
        is(1),  -- entity is an intermediate system
                -- entity is an end system and does
        es(2)  -- not forward PDUs
    }
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The indication of whether this entity is active
         as an intermediate or end system.  Only
         intermediate systems will forward PDUs onward that
         are not addressed to them."
    ::= { clnp 1 }

clnpDefaultLifeTime OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The default value inserted into the Lifetime
         field of the CLNP PDU header of PDUs sourced by
         this entity."
    ::= { clnp 2 }

clnpInReceives OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The total number of input PDUs received from all
         connected network interfaces running CLNP,
         including errors."
    ::= { clnp 3 }

clnpInHdrErrors OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The number of input PDUs discarded due to errors
         in the CLNP header, including bad checksums,
         version mismatch, lifetime exceeded, errors
         discovered in processing options, etc."
    ::= { clnp 4 }

```

## clnpInAddrErrors OBJECT-TYPE

SYNTAX Counter  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION

"The number of input PDUs discarded because the NSAP address in the CLNP header's destination field was not a valid NSAP to be received at this entity. This count includes addresses not understood. For end systems, this is a count of PDUs which arrived with a destination NSAP which was not local."

::= { clnp 5 }

## clnpForwPDUs OBJECT-TYPE

SYNTAX Counter  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION

"The number of input PDUs for which this entity was not the final destination and which an attempt was made to forward them onward."

::= { clnp 6 }

## clnpInUnknownNLPs OBJECT-TYPE

SYNTAX Counter  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION

"The number of locally-addressed PDUs successfully received but discarded because the network layer protocol was unknown or unsupported (e.g., not CLNP or ES-IS)."

::= { clnp 7 }

## clnpInUnknownULPs OBJECT-TYPE

SYNTAX Counter  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION

"The number of locally-addressed PDUs successfully received but discarded because the upper layer protocol was unknown or unsupported (e.g., not TP4)."

::= { clnp 8 }

## clnpInDiscards OBJECT-TYPE

SYNTAX Counter

```
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The number of input CLNP PDUs for which no
    problems were encountered to prevent their
    continued processing, but were discarded (e.g.,
    for lack of buffer space). Note that this counter
    does not include any PDUs discarded while awaiting
    re-assembly."
 ::= { clnp 9 }

clnpInDelivers OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The total number of input PDUs successfully
    delivered to the CLNS transport user."
 ::= { clnp 10 }

clnpOutRequests OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The total number of CLNP PDUs which local CLNS
    user protocols supplied to CLNP for transmission
    requests. This counter does not include any PDUs
    counted in clnpForwPDUs."
 ::= { clnp 11 }

clnpOutDiscards OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The number of output CLNP PDUs for which no other
    problem was encountered to prevent their
    transmission but were discarded (e.g., for lack of
    buffer space). Note this counter includes PDUs
    counted in clnpForwPDUs."
 ::= { clnp 12 }

clnpOutNoRoutes OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
```

```

        "The number of CLNP PDUs discarded because no
        route could be found to transmit them to their
        destination. This counter includes any PDUs
        counted in clnpForwPDUs."
 ::= { clnp 13 }

clnpReasmTimeout OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The maximum number of seconds which received
        segments are held while they are awaiting
        reassembly at this entity."
 ::= { clnp 14 }

clnpReasmReqds OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The number of CLNP segments received which needed
        to be reassembled at this entity."
 ::= { clnp 15 }

clnpReasmOKs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The number of CLNP PDUs successfully re-assembled
        at this entity."
 ::= { clnp 16 }

clnpReasmFails OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The number of failures detected by the CLNP
        reassembly algorithm (for any reason: timed out,
        buffer size, etc).".
 ::= { clnp 17 }

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

DESCRIPTION  
    "The number of CLNP PDUs that have been  
    successfully segmented at this entity."  
 ::= { clnp 18 }

clnpSegFails OBJECT-TYPE  
SYNTAX Counter  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The number of CLNP PDUs that have been discarded  
    because they needed to be fragmented at this  
    entity but could not."  
 ::= { clnp 19 }

clnpSegCreates OBJECT-TYPE  
SYNTAX Counter  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The number of CLNP PDU segments that have been  
    generated as a result of segmentation at this  
    entity."  
 ::= { clnp 20 }

clnpInOpts OBJECT-TYPE  
SYNTAX Counter  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The number of CLNP PDU segments that have been  
    input with options at this entity."  
 ::= { clnp 25 }

clnpOutOpts OBJECT-TYPE  
SYNTAX Counter  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
    "The number of CLNP PDU segments that have been  
    generated with options by this entity."  
 ::= { clnp 26 }

clnpRoutingDiscards OBJECT-TYPE  
SYNTAX Counter  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION

```

        "The number of routing entries which were chosen
        to be discarded even though they are valid. One
        possible reason for discarding such an entry could
        be to free-up buffer space for other routing
        entries."
 ::= { clnp 27 }

-- the CLNP Interfaces table

-- The CLNP interfaces table contains information on the
-- entity's interfaces which are running the CLNP.

clnpAddrTable OBJECT-TYPE
    SYNTAX SEQUENCE OF ClnpAddrEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "The table of addressing information relevant to
        this entity's CLNP addresses. "
    ::= { clnp 21 }

clnpAddrEntry OBJECT-TYPE
    SYNTAX ClnpAddrEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "The addressing information for one of this
        entity's CLNP addresses."
    INDEX { clnpAdEntAddr }
    ::= { clnpAddrTable 1 }

ClnpAddrEntry ::=
    SEQUENCE {
        clnpAdEntAddr
            ClnpAddress,
        clnpAdEntIfIndex
            INTEGER,
        clnpAdEntReasmMaxSize
            INTEGER (0..65535)
    }

clnpAdEntAddr OBJECT-TYPE
    SYNTAX ClnpAddress
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The CLNP address to which this entry's addressing

```

```

        information pertains."
 ::= { clnpAddrEntry 1 }

clnpAdEntIfIndex OBJECT-TYPE
SYNTAX  INTEGER
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "The index value which uniquely identifies the
        interface to which this entry is applicable.  The
        interface identified by a particular value of this
        index is the same interface as identified by the
        same value of ifIndex."
 ::= { clnpAddrEntry 2 }

clnpAdEntReasmMaxSize OBJECT-TYPE
SYNTAX  INTEGER (0..65535)
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
        "The size of the largest CLNP PDU which this
        entity can re-assemble from incoming CLNP
        segmented PDUs received on this interface."
 ::= { clnpAddrEntry 3 }

-- The CLNP Routing table

-- The CLNP routing table contains an entry for each route
-- known to the entity.

clnpRoutingTable OBJECT-TYPE
SYNTAX  SEQUENCE OF ClnpRouteEntry
ACCESS  not-accessible
STATUS  mandatory
DESCRIPTION
        "This entity's CLNP routing table."
 ::= { clnp 22 }

clnpRouteEntry OBJECT-TYPE
SYNTAX  ClnpRouteEntry
ACCESS  not-accessible
STATUS  mandatory
DESCRIPTION
        "A route to a particular destination."
INDEX   { clnpRouteDest }
 ::= { clnpRoutingTable 1 }

```

```

ClnpRouteEntry ::=
  SEQUENCE {
    clnpRouteDest
      ClnpAddress,
    clnpRouteIfIndex
      INTEGER,
    clnpRouteMetric1
      INTEGER,
    clnpRouteMetric2
      INTEGER,
    clnpRouteMetric3
      INTEGER,
    clnpRouteMetric4
      INTEGER,
    clnpRouteNextHop
      ClnpAddress,
    clnpRouteType
      INTEGER,
    clnpRouteProto
      INTEGER,
    clnpRouteAge
      INTEGER,
    clnpRouteMetric5
      INTEGER,
    clnpRouteInfo
      OBJECT IDENTIFIER
  }

```

```

clnpRouteDest OBJECT-TYPE
  SYNTAX ClnpAddress
  ACCESS read-write
  STATUS mandatory
  DESCRIPTION
    "The destination CLNP address of this route."
  ::= { clnpRouteEntry 1 }

```

```

clnpRouteIfIndex OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-write
  STATUS mandatory
  DESCRIPTION
    "The index value which uniquely identifies the
    local interface through which the next hop of this
    route should be reached. The interface identified
    by a particular value of this index is the same as
    identified by the same value of ifIndex."
  ::= { clnpRouteEntry 2 }

```

```
clnpRouteMetric1 OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The primary routing metric for this route.  The
        semantics of this metric are determined by the
        routing-protocol specified in the route's
        clnpRouteProto value.  If this metric is not used,
        its value should be set to -1."
    ::= { clnpRouteEntry 3 }

clnpRouteMetric2 OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "An alternate routing metric for this route.  The
        semantics of this metric are determined by the
        routing-protocol specified in the route's
        clnpRouteProto value.  If this metric is not used,
        its value should be set to -1."
    ::= { clnpRouteEntry 4 }

clnpRouteMetric3 OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "An alternate routing metric for this route.  The
        semantics of this metric are determined by the
        routing-protocol specified in the route's
        clnpRouteProto value.  If this metric is not used,
        its value should be set to -1."
    ::= { clnpRouteEntry 5 }

clnpRouteMetric4 OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "An alternate routing metric for this route.  The
        semantics of this metric are determined by the
        routing-protocol specified in the route's
        clnpRouteProto value.  If this metric is not used,
        its value should be set to -1."
    ::= { clnpRouteEntry 6 }
```

```

clnpRouteNextHop OBJECT-TYPE
    SYNTAX  ClnpAddress
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The CLNP address of the next hop of this route."
    ::= { clnpRouteEntry 7 }

clnpRouteType OBJECT-TYPE
    SYNTAX  INTEGER {
        other(1),          -- none of the following
                          --
        invalid(2),       -- an invalidated route
                          --
        direct(3),        -- route to directly
                          -- connected (sub-)network
                          --
        remote(4)         -- route to a non-local
                          -- host/network/sub-network
    }
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The type of route.

        Setting this object to the value invalid(2) has
        the effect of invalidating the corresponding entry
        in the clnpRoutingTable. That is, it effectively
        disassociates the destination identified with said
        entry from the route identified with said entry.
        It is an implementation-specific matter as to
        whether the agent removes an invalidated entry
        from the table. Accordingly, management stations
        must be prepared to receive tabular information
        from agents that corresponds to entries not
        currently in use. Proper interpretation of such
        entries requires examination of the relevant
        clnpRouteType object."
    ::= { clnpRouteEntry 8 }

clnpRouteProto OBJECT-TYPE
    SYNTAX  INTEGER {
        other(1),          -- none of the following
                          --
                          -- non-protocol information
                          -- e.g., manually
        local(2),         -- configured entries
    }

```

```

                                -- set via a network
                                -- management protocol
                                -- similar to ipRouteProto but
                                -- omits several IP-specific
                                -- protocols

                                is-is(9),
                                ciscoIgrp(11),
                                bbnSpfIgp(12),
                                ospf(13),
                                bgp(14)
                                }
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The routing mechanism via which this route was
    learned. Inclusion of values for gateway routing
    protocols is not intended to imply that hosts
    should support those protocols."
 ::= { clnpRouteEntry 9 }

clnpRouteAge OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-write
STATUS mandatory
DESCRIPTION
    "The number of seconds since this route was last
    updated or otherwise determined to be correct.
    Note that no semantics of 'too old' can be implied
    except through knowledge of the routing protocol
    by which the route was learned."
 ::= { clnpRouteEntry 10 }

clnpRouteMetric5 OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-write
STATUS mandatory
DESCRIPTION
    "An alternate routing metric for this route. The
    semantics of this metric are determined by the
    routing-protocol specified in the route's
    clnpRouteProto value. If this metric is not used,
    its value should be set to -1."
 ::= { clnpRouteEntry 11 }

clnpRouteInfo OBJECT-TYPE
SYNTAX OBJECT IDENTIFIER

```

```

ACCESS    read-only
STATUS    mandatory
DESCRIPTION
    "A reference to MIB definitions specific to the
    particular routing protocol which is responsible
    for this route, as determined by the value
    specified in the route's clnpRouteProto value.  If
    this information is not present, its value should
    be set to the OBJECT IDENTIFIER { 0 0 }, which is
    a syntatically valid object identifier, and any
    conformant implementation of ASN.1 and BER must be
    able to generate and recognize this value."
 ::= { clnpRouteEntry 12 }

-- the CLNP Address Translation table

-- The Address Translation tables contain the CLNP address
-- to physical address equivalences.  Some interfaces do not
-- use translation tables for determining address
-- equivalences; if all interfaces are of this type, then the
-- Address Translation table is empty, i.e., has zero
-- entries.

clnpNetToMediaTable OBJECT-TYPE
    SYNTAX    SEQUENCE OF ClnpNetToMediaEntry
    ACCESS    not-accessible
    STATUS    mandatory
    DESCRIPTION
        "The CLNP Address Translation table used for
        mapping from CLNP addresses to physical
        addresses."
    ::= { clnp 23 }

clnpNetToMediaEntry OBJECT-TYPE
    SYNTAX    ClnpNetToMediaEntry
    ACCESS    not-accessible
    STATUS    mandatory
    DESCRIPTION
        "Each entry contains one CLNP address to
        'physical' address equivalence."
    INDEX    { clnpNetToMediaIfIndex, clnpNetToMediaNetAddress }
    ::= { clnpNetToMediaTable 1 }

ClnpNetToMediaEntry ::=
    SEQUENCE {
        clnpNetToMediaIfIndex
            INTEGER,

```

```

        clnpNetToMediaPhysAddress
            PhysAddress,
        clnpNetToMediaNetAddress
            ClnpAddress,
        clnpNetToMediaType
            INTEGER,
        clnpNetToMediaAge
            INTEGER,
        clnpNetToMediaHoldTime
            INTEGER
    }

clnpNetToMediaIfIndex OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The interface on which this entry's equivalence
         is effective. The interface identified by a
         particular value of this index is the same
         interface as identified by the same value of
         ifIndex."
    ::= { clnpNetToMediaEntry 1 }

clnpNetToMediaPhysAddress OBJECT-TYPE
    SYNTAX  PhysAddress
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The media-dependent 'physical' address."
    ::= { clnpNetToMediaEntry 2 }

clnpNetToMediaNetAddress OBJECT-TYPE
    SYNTAX  ClnpAddress
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The CLNP address corresponding to the media-
         dependent 'physical' address."
    ::= { clnpNetToMediaEntry 3 }

clnpNetToMediaType OBJECT-TYPE
    SYNTAX  INTEGER {
        other(1),          -- none of the following
        invalid(2),       -- an invalidated mapping
        dynamic(3),
        static(4)
    }

```

ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION

"The type of mapping.

Setting this object to the value invalid(2) has the effect of invalidating the corresponding entry in the clnpNetToMediaTable. That is, it effectively disassociates the interface identified with said entry from the mapping identified with said entry. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries not currently in use. Proper interpretation of such entries requires examination of the relevant clnpNetToMediaType object."

::= { clnpNetToMediaEntry 4 }

clnpNetToMediaAge OBJECT-TYPE

SYNTAX INTEGER  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION

"The number of seconds since this entry was last updated or otherwise determined to be correct. Note that no semantics of 'too old' can be implied except through knowledge of the type of entry."

::= { clnpNetToMediaEntry 5 }

clnpNetToMediaHoldTime OBJECT-TYPE

SYNTAX INTEGER  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION

"The time in seconds this entry will be valid. Static entries should always report this field as -1."

::= { clnpNetToMediaEntry 6 }

clnpMediaToNetTable OBJECT-TYPE

SYNTAX SEQUENCE OF ClnpMediaToNetEntry  
 ACCESS not-accessible  
 STATUS mandatory  
 DESCRIPTION

"The CLNP Address Translation table used for

```

        mapping from physical addresses to CLNP
        addresses."
 ::= { clnp 24 }

clnpMediaToNetEntry OBJECT-TYPE
SYNTAX ClnpMediaToNetEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "Each entry contains on ClnpAddress to 'physical'
    address equivalence."
INDEX { clnpMediaToNetIfIndex, clnpMediaToNetPhysAddress }
 ::= { clnpMediaToNetTable 1 }

ClnpMediaToNetEntry ::=
SEQUENCE {
    clnpMediaToNetIfIndex
        INTEGER,
    clnpMediaToNetNetAddress
        ClnpAddress,
    clnpMediaToNetPhysAddress
        PhysAddress,
    clnpMediaToNetType
        INTEGER,
    clnpMediaToNetAge
        INTEGER,
    clnpMediaToNetHoldTime
        INTEGER
}

clnpMediaToNetIfIndex OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-write
STATUS mandatory
DESCRIPTION
    "The interface on which this entry's equivalence
    is effective. The interface identified by a
    particular value of this index is the same
    interface as identified by the same value of
    ifIndex."
 ::= { clnpMediaToNetEntry 1 }

clnpMediaToNetAddress OBJECT-TYPE
SYNTAX ClnpAddress
ACCESS read-write
STATUS mandatory
DESCRIPTION
    "The ClnpAddress corresponding to the media-
```

```

        dependent 'physical' address."
 ::= { clnpMediaToNetEntry 2 }

```

clnpMediaToNetPhysAddress OBJECT-TYPE

```

SYNTAX  PhysAddress
ACCESS  read-write
STATUS  mandatory
DESCRIPTION
        "The media-dependent 'physical' address."
 ::= { clnpMediaToNetEntry 3 }

```

clnpMediaToNetType OBJECT-TYPE

```

SYNTAX  INTEGER {
        other(1),          -- none of the following
        invalid(2),       -- an invalidated mapping
        dynamic(3),
        static(4)
    }

```

```

ACCESS  read-write
STATUS  mandatory
DESCRIPTION
        "The type of mapping.

```

Setting this object to the value invalid(2) has the effect of invalidating the corresponding entry in the clnpMediaToNetTable. That is, it effectively disassociates the interface identified with said entry from the mapping identified with said entry. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries not currently in use. Proper interpretation of such entries requires examination of the relevant clnpMediaToNetType object."

```

 ::= { clnpMediaToNetEntry 4 }

```

clnpMediaToNetAge OBJECT-TYPE

```

SYNTAX  INTEGER
ACCESS  read-write
STATUS  mandatory
DESCRIPTION
        "The number of seconds since this entry was last
        updated or otherwise determined to be correct.
        Note that no semantics of 'too old' can be implied
        except through knowledge of the type of entry."

```

```
 ::= { clnpMediaToNetEntry 5 }

clnpMediaToNetHoldTime OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "The time in seconds this entry will be valid.
        Static entries should always report this field as
        -1."
    ::= { clnpMediaToNetEntry 6 }

-- the CLNP Error group

-- Implementation of this group is recommended for all
-- systems which implement the CLNP Error protocol.

clnpInErrors OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The number of CLNP Error PDUs received by this
        entity."
    ::= { error 1 }

clnpOutErrors OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The number of CLNP Error PDUs sent by this
        entity."
    ::= { error 2 }

clnpInErrUnspecs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "The number of unspecified CLNP Error PDUs
        received by this entity."
    ::= { error 3 }

clnpInErrProcs OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
```

```
STATUS mandatory
DESCRIPTION
    "The number of protocol procedure CLNP Error PDUs
    received by this entity."
 ::= { error 4 }

clnpInErrCksums OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The number of checksum CLNP Error PDUs received
    by this entity."
 ::= { error 5 }

clnpInErrCongests OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The number of congestion drop CLNP Error PDUs
    received by this entity."
 ::= { error 6 }

clnpInErrHdrs OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The number of header syntax CLNP Error PDUs
    received by this entity."
 ::= { error 7 }

clnpInErrSegs OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The number of segmentation disallowed CLNP Error
    PDUs received by this entity."
 ::= { error 8 }

clnpInErrIncomps OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The number of incomplete PDU CLNP Error PDUs
```

```
        received by this entity."
 ::= { error 9 }

clnpInErrDups OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of duplicate option CLNP Error PDUs
         received by this entity."
 ::= { error 10 }

clnpInErrUnreachDsts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of unreachable destination CLNP Error
         PDUs received by this entity."
 ::= { error 11 }

clnpInErrUnknownDsts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of unknown destination CLNP Error PDUs
         received by this entity."
 ::= { error 12 }

clnpInErrSRUnspecs OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of unspecified source route CLNP Error
         PDUs received by this entity."
 ::= { error 13 }

clnpInErrSRSyntaxes OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of source route syntax CLNP Error PDUs
         received by this entity."
 ::= { error 14 }
```

```
clnpInErrSRUnkAdrrs OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of source route unknown address CLNP
        Error PDUs received by this entity."
    ::= { error 15 }

clnpInErrSRBadPaths OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of source route bad path CLNP Error
        PDUs received by this entity."
    ::= { error 16 }

clnpInErrHops OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of hop count exceeded CLNP Error PDUs
        received by this entity."
    ::= { error 17 }

clnpInErrHopReassms OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of hop count exceeded while
        reassembling CLNP Error PDUs received by this
        entity."
    ::= { error 18 }

clnpInErrUnsOptions OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of unsupported option CLNP Error PDUs
        received by this entity."
    ::= { error 19 }

clnpInErrUnsVersions OBJECT-TYPE
    SYNTAX Counter
```

```
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The number of version mismatch CLNP Error PDUs
    received by this entity."
 ::= { error 20 }

clnpInErrUnsSecurities OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The number of unsupported security option CLNP
    Error PDUs received by this entity."
 ::= { error 21 }

clnpInErrUnsSRs OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The number of unsupported source route option
    CLNP Error PDUs received by this entity."
 ::= { error 22 }

clnpInErrUnsRRs OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The number of unsupported record route option
    CLNP Error PDUs received by this entity."
 ::= { error 23 }

clnpInErrInterferences OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The number of reassembly interference CLNP Error
    PDUs received by this entity."
 ::= { error 24 }

clnpOutErrUnspecs OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
```

```
        "The number of unspecified CLNP Error PDUs sent by
        this entity."
 ::= { error 25 }

clnpOutErrProcs OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of protocol procedure CLNP Error PDUs
        sent by this entity."
 ::= { error 26 }

clnpOutErrCksums OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of checksum CLNP Error PDUs sent by
        this entity."
 ::= { error 27 }

clnpOutErrCongests OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of congestion drop CLNP Error PDUs
        sent by this entity."
 ::= { error 28 }

clnpOutErrHdrs OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of header syntax CLNP Error PDUs sent
        by this entity."
 ::= { error 29 }

clnpOutErrSegs OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of segmentation disallowed CLNP Error
        PDUs sent by this entity."
 ::= { error 30 }
```

```
clnpOutErrIncomps OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of incomplete PDU CLNP Error PDUs sent
         by this entity."
    ::= { error 31 }

clnpOutErrDups OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of duplicate option CLNP Error PDUs
         sent by this entity."
    ::= { error 32 }

clnpOutErrUnreachDsts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of unreachable destination CLNP Error
         PDUs sent by this entity."
    ::= { error 33 }

clnpOutErrUnknownDsts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of unknown destination CLNP Error PDUs
         sent by this entity."
    ::= { error 34 }

clnpOutErrSRUnspecs OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of unspecified source route CLNP Error
         PDUs sent by this entity."
    ::= { error 35 }

clnpOutErrSRSyntaxes OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
```

```
STATUS mandatory
DESCRIPTION
    "The number of source route syntax CLNP Error PDUs
    sent by this entity."
 ::= { error 36 }

clnpOutErrSRUnkAdrrs OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The number of source route unknown address CLNP
    Error PDUs sent by this entity."
 ::= { error 37 }

clnpOutErrSRBadPaths OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The number of source route bad path CLNP Error
    PDUs sent by this entity."
 ::= { error 38 }

clnpOutErrHops OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The number of hop count exceeded CLNP Error PDUs
    sent by this entity."
 ::= { error 39 }

clnpOutErrHopReassms OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The number of hop count exceeded while
    reassembling CLNP Error PDUs sent by this entity."
 ::= { error 40 }

clnpOutErrUnsOptions OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The number of unsupported option CLNP Error PDUs
```

```
        sent by this entity."
 ::= { error 41 }

clnpOutErrUnsVersions OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of version mismatch CLNP Error PDUs
        sent by this entity."
 ::= { error 42 }

clnpOutErrUnsSecurities OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of unsupported security option CLNP
        Error PDUs sent by this entity."
 ::= { error 43 }

clnpOutErrUnsSRs OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of unsupported source route option
        CLNP Error PDUs sent by this entity."
 ::= { error 44 }

clnpOutErrUnsRRs OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of unsupported record route option
        CLNP Error PDUs sent by this entity."
 ::= { error 45 }

clnpOutErrInterferences OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of reassembly interference CLNP Error
        PDUs sent by this entity."
 ::= { error 46 }
```

```
-- the ES-IS group

-- Implementation of this group is recommended for all
-- systems which implement the End-System to Intermediate
-- System protocol.

esisESHins OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of ESH PDUs received by this entity."
    ::= { es-is 1 }

esisESHouts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of ESH PDUs sent by this entity."
    ::= { es-is 2 }

esisISHins OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of ISH PDUs received by this entity."
    ::= { es-is 3 }

esisISHouts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of ISH PDUs sent by this entity."
    ::= { es-is 4 }

esisRDUins OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The number of RDU PDUs received by this entity."
    ::= { es-is 5 }

esisRDUouts OBJECT-TYPE
    SYNTAX Counter
```

```
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The number of RDU PDUs sent by this entity."
 ::= { es-is 6 }
```

END

## 5. References

- [1] Cerf, V., "IAB Recommendations for the Development of Internet Network Management Standards", RFC 1052, IAB, 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 and 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 and Performance Systems International, May 1990.
- [5] Case, J., M. Fedor, M. Schoffstall, and J. Davin, "The Simple Network Management Protocol", RFC 1157, University of Tennessee at Knoxville, Performance Systems International, Performance Systems International, and the 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, Hughes LAN Systems, Inc., 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,

Inc., March 1991.

- [10] Information processing systems - Data Communications - Protocol for providing the Connectionless-mode Network Service and Provision of Underlying Service, International Organization for Standardization, International Standard 8473, May 1987.
- [11] End System to Intermediate System Routing Exchange Protocol for Use in Conjunction with the Protocol for the Provision of the Connectionless-mode Network Service (ISO 8473), International Draft Proposal 9542.
- [12] Kille, S., "A String Encoding of Presentation Address", Research Note RN/89/14, Department of Computer Science, University College London, February 1989.

## 6. Security Considerations

Security issues are not discussed in this memo.

## 7. Author's Address:

Greg Satz  
cisco Systems, Inc.  
1350 Willow Road  
Menlo Park, CA 94025

Phone: (415) 326-1941

Email: Satz@CISCO.COM