Chapter 4

SNMPv1 Network Management: Organization and Information Models
Objectives

- Organization Model
  - 2- and 3-tier models
  - Manager and agent
- Management messages
- Structure of management information, SMI
- Object type and instance
- Scalar and aggregate managed objects
- Management information base, MIB
- NMS physical and virtual databases
- IETF MIB-2 standard
Managed LAN

Figure 4.1 Managed LAN Network

Notes

• NMS on subnet 192.168.252.1 manages the router and the hubs on subnet 172.16.46.1 across the backbone network
Managed Hub:  
System Information

Title: System Information: 172.16.46.2  
Name or IP Address: 172.16.46.2

System Name:  
System Description: 3Com LinkBuilder FMS, SW version: 3.02  
System Contact:  
System Location:  
System Object ID: iso.org.dod.internet.private.enterprises.43.1.8.5  
System Up Time: (2475380437) 286 days, 12:03:24.37

Figure 4.2(a) System Information on 172.16.46.2 Hub

Notes

• Information obtained querying the hub
• Data truly reflects what is stored in the hub
Managed Hub:

System Information

Title:  System Information:  172.16.46.3
Name or IP Address:  172.16.46.3

System Name:
System Description:  3Com LinkBuilder FMS, SW version:3.12
System Contact:
System Location:
System Object ID:     iso.org.dod.internet.private.enterprises.43.1.8.5
System Up Time:      (3146735182)  364 days, 4:55:51.82

Figure 4.2(b)  System Information on 172.16.46.3 Hub
Managed Router: System Information

Title: System Information: router1.gatech.edu
Name or IP Address: 172.16.46.1

System Name : router1.gatech.edu
System Description : Cisco Internetwork Operating System Software
                    : IOS (tm) 7000 Software (C7000-JS-M), Version
                    : 11.2(6),RELEASE SOFTWARE (ge1)
                    : Copyright (c) 1986-1997 by Cisco Systems, Inc.
                    : Compiled Tue 06-May-97 19:11 by kuong

System Contact : 
System Location : 
cisco 7000
System Up Time : (315131795) 36 days, 11:21:57.95

Figure 4.2(c) System Information on Router
### Managed Hub: Port Addresses

<table>
<thead>
<tr>
<th>Index</th>
<th>Interface</th>
<th>IP address</th>
<th>Network Mask</th>
<th>Network Address</th>
<th>Link Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3Com</td>
<td>172.16.46.2</td>
<td>255.255.255.0</td>
<td>172.16 46.0</td>
<td>0x08004E07C25C</td>
</tr>
<tr>
<td>2</td>
<td>3Com</td>
<td>192.168.252.1</td>
<td>255.255.255.0</td>
<td>192.168.252.0</td>
<td>&lt;none&gt;</td>
</tr>
</tbody>
</table>

**Notes**

- Information acquired by the NMS on hub interfaces
- Index refers to the interface on the hub
- Link address is the MAC address
- The second row data is a serial link
Managed Router: Port Addresses

<table>
<thead>
<tr>
<th>Index</th>
<th>Interface</th>
<th>IP address</th>
<th>Network Mask</th>
<th>Network Address</th>
<th>Link Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>LEC.1.0</td>
<td>192.168.3.1</td>
<td>255.255.255.0</td>
<td>192.168.3.0</td>
<td>0x00000C3920B4</td>
</tr>
<tr>
<td>25</td>
<td>LEC.3.9</td>
<td>192.168.252.1</td>
<td>255.255.255.0</td>
<td>192.168.252.0</td>
<td>0x00000C3920B4</td>
</tr>
<tr>
<td>13</td>
<td>Ethernet2/0</td>
<td>172.16.46.1</td>
<td>255.255.255.0</td>
<td>172.16.46.1</td>
<td>0x00000C3920AC</td>
</tr>
<tr>
<td>16</td>
<td>Ethernet2/3</td>
<td>172.16.49.1</td>
<td>255.255.255.0</td>
<td>172.16.49.0</td>
<td>0x00000C3920AF</td>
</tr>
<tr>
<td>17</td>
<td>Ethernet2/4</td>
<td>172.16.52.1</td>
<td>255.255.255.0</td>
<td>172.16.52.0</td>
<td>0x00000C3920B0</td>
</tr>
<tr>
<td>9</td>
<td>Ethernet1/2</td>
<td>172.16.55.1</td>
<td>255.255.255.0</td>
<td>172.16.55.0</td>
<td>0x00000C3920A6</td>
</tr>
<tr>
<td>2</td>
<td>Ethernet 0/1</td>
<td>172.16.56.1</td>
<td>255.255.255.0</td>
<td>172.16.56.0</td>
<td>0x00000C39209D</td>
</tr>
<tr>
<td>15</td>
<td>Ethernet2/2</td>
<td>172.16.57.1</td>
<td>255.255.255.0</td>
<td>172.16.57.0</td>
<td>0x00000C3920AE</td>
</tr>
<tr>
<td>8</td>
<td>Ethernet1/1</td>
<td>172.16.58.1</td>
<td>255.255.255.0</td>
<td>172.16.58.0</td>
<td>0x00000C3920A5</td>
</tr>
<tr>
<td>14</td>
<td>Ethernet2/1</td>
<td>172.16.60.1</td>
<td>255.255.255.0</td>
<td>172.16.60.0</td>
<td>0x00000C3920AD</td>
</tr>
</tbody>
</table>

**Notes**

- Information acquired by NMS on the router interfaces
- Index refers to the interface on the router
- LEC is the LAN emulation card
- Ethernet 2/0 interface refers to the interface card 2 and port 0 in that card
SNMPv1 & SNMPv2
Documents

Figure 4.4 SNMP Document Evolution
SNMP Model

• Organization Model
  • Relationship between network element, agent, and manager
  • Hierarchical architecture
• Information Model
  • Uses ASN.1 syntax
  • SMI (Structure of Management Information)
  • MIB (Management Information Base)
• Communication Model
  • Transfer syntax
  • SNMP over TCP/IP
  • Communication services addressed by messages
Two-Tier Organization Model

(a) One Manager-One Agent Model  
(b) Multiple Managers-One Agent Model

Figure 4.5 Two-Tier Organization Model

Notes

• Any host that could query an agent is a manager.
Three-Tier Organization Model: RMON

Notes

- Managed object comprises network element and management agent
- RMON acts as an agent and a manager
- RMON (Remote Monitoring) gathers data from MO, analyses the data, and stores the data
- Communicates the statistics to the manager
Three-Tier Organization Model: Proxy Server

SNMP Manager

Proxy Server

Non-SNMP Managed Objects

SNMP Managed Objects

Figure 4.7 Proxy Server Organization Model

Notes

• Proxy server converts non-SNMP data from non-SNMP objects to SNMP compatible objects and messages
System Architecture

SNMP Manager

SNMP Agent

Management Data

SNMP Manager Application

SNMP Agent Application

SNMP

UDP

IP

DLC

PHY

SNMP

UDP

IP

DLC

PHY

Physical Medium

Figure 4.9  SNMP Network Management Architecture

Notes

• Messages between manager and agent
• Direction of messages - 3 from manager and 2 from agent
SNMP Messages

• Get-Request
  • Sent by manager requesting data from agent
• Get-Next-Request
  • Sent by manager requesting data on the next MO to the one specified
• Set-Request
  • Initializes or changes the value of MO
• Get-Response
  • Agent responds with data for get and set requests from the manager
• Trap
  • Alarm generated by an agent
Managed Object

Object
  ▼
  Object Type
    ▼
    Name: OBJECT IDENTIFIER
    Syntax: ASN.1
    Encoding: BER

Object Instance

Figure 4.10  Managed Object: Type and Instance

Notes

- Object type and data type are synonymous
- Object identifier is data type, not instance
Managed Object: Multiple Instances

Figure 4.11 Managed Object: Type with Multiple Instances

Notes

• All 3 Com hubs of the same version have identical identifier; they are distinguished by the IP address.
• Each IP address is an instance of the object.
Name

Uniquely defined by

• DESCRIPTOR AND
• OBJECT IDENTIFIER

internet OBJECT IDENTIFIER ::= {iso org(3) dod(6) 1 }.
Internet Subnodes

Figure 4.13 Subnodes under Internet Node in SNMPv1

Notes

- directory
- mgmt
- experimental
- private

OBJECT IDENTIFIER ::= {internet 1}
OBJECT IDENTIFIER ::= {internet 2}
OBJECT IDENTIFIER ::= {internet 3}
OBJECT IDENTIFIER ::= {internet 4}
Private MIB Example

![Diagram of a private MIB example with nodes for internet, private, enterprises, cisco, hp, 3Com, and Cabletron.]

Figure 4.14 Private Subtree for Commercial Vendors

Notes

- *private* MIB intended for vendor equipment
- IANA (Internet Assigned Numbers Authority) assigns identifiers
SNMP ASN.1 Data Type

Figure 4.15  SNMP ASN.1 Data Type
## Primitive Data Types

<table>
<thead>
<tr>
<th>Structure</th>
<th>Data Type</th>
<th>Comments</th>
</tr>
</thead>
</table>
| Primitive types   | INTEGER         | Subtype INTEGER (n1..nN)  
Special case: Enumerated INTEGER type                                      |
|                   | OCTET STRING    | 8-bit bytes binary and textual data  
Subtypes can be specified by either range or fixed                       |
|                   | OBJECT IDENTIFIER | Object position in MIB                                                  |
|                   | NULL            | Placeholder                                                              |

## Notes

- *get-request* message has NULL for value fields and *get-response* from agent has the values filled in subtype:
  - INTEGER (0..255)
  - OCTET STRING (SIZE 0..255)
  - OCTET STRING (SIZE 8)
Enumerated

• Special case of INTEGER data type

```plaintext
error-status INTEGER {
    noError(0)
    tooBig(1)
    genErr(5)
    authorizationError(16)
}
```

Notes

• noError NULL by convention
Defined or Application Data Type

<table>
<thead>
<tr>
<th>Defined Types</th>
<th>NetworkAddress</th>
<th>Not used</th>
</tr>
</thead>
<tbody>
<tr>
<td>IpAddress</td>
<td></td>
<td>Dotted decimal IP address</td>
</tr>
<tr>
<td>Counter</td>
<td></td>
<td>Wrap-around, non-negative integer, monotonically increasing, max 2^32 -1</td>
</tr>
<tr>
<td>Gauge</td>
<td></td>
<td>Capped, non-negative integer, increase or decrease</td>
</tr>
<tr>
<td>TimeTicks</td>
<td></td>
<td>Non-negative integer in hundredths of second units</td>
</tr>
<tr>
<td>Opaque</td>
<td></td>
<td>Application-wide arbitrary ASN.1 syntax, double wrapped OCTET STRING</td>
</tr>
</tbody>
</table>

Notes

- Defined data types are simple or base types
- Opaque is used to create data types based on previously defined data types
Constructor or Structured Data Type: SEQUENCE

• List maker

SEQUENCE { <type1>, <type2>,..., <typeN> }

<table>
<thead>
<tr>
<th>Object</th>
<th>OBJECT IDENTIFIER</th>
<th>ObjectSyntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ipAdEntAddr</td>
<td>{ipAddrEntry 1}</td>
<td>IpAddress</td>
</tr>
<tr>
<td>2 ipAdEntIfIndex</td>
<td>{ipAddrEntry 2}</td>
<td>INTEGER</td>
</tr>
<tr>
<td>3 ipAdEntNetMask</td>
<td>{ipAddrEntry 3}</td>
<td>IpAddress</td>
</tr>
<tr>
<td>4 ipAdEntBcastAddr</td>
<td>{ipAddrEntry 4}</td>
<td>INTEGER</td>
</tr>
<tr>
<td>5 ipAdEntReasmMaxSize</td>
<td>{ipAddrEntry 5}</td>
<td>INTEGER</td>
</tr>
<tr>
<td>6 ipAddrEntry</td>
<td>{ipAddrTable 1}</td>
<td>SEQUENCE</td>
</tr>
</tbody>
</table>

List: IpAddrEntry ::= 

SEQUENCE {
    ipAdEntAddr       IpAddress
    ipAdEntIfIndex    INTEGER
    ipAdEntNetMask    IpAddress
    ipAdEntBcastAddr  INTEGER
    ipAdEntReasmMaxSize INTEGER (0..65535)
} 

Managed Object IpAddrEntry as a list
Constructor or Structured Data Type: SEQUENCE OF

SEQUENCE OF <entry>
where <entry> is a list constructor

<table>
<thead>
<tr>
<th>Object Name</th>
<th>OBJECT IDENTIFIER</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 ipAddrTable</td>
<td>{ip 20}</td>
<td>SEQUENCE OF</td>
</tr>
</tbody>
</table>

Table: IpAddrTable ::= 
  SEQUENCE OF IpAddrEntry

Managed Object ipAddrTable as a table
### SEQUENCE OF Example

Title: System Information : router1.gatech.edu  
Name or IP Address: 172.16252.1

<table>
<thead>
<tr>
<th>Index</th>
<th>Interface</th>
<th>IP address</th>
<th>Network Mask</th>
<th>Network Address</th>
<th>Link Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>LEC.1.0</td>
<td>192.168.3.1</td>
<td>255.255.255.0</td>
<td>192.168.3.0</td>
<td>0x00000C3920B4</td>
</tr>
<tr>
<td>25</td>
<td>LEC.3.9</td>
<td>192.168.252.1</td>
<td>255.255.255.0</td>
<td>192.168.252.0</td>
<td>0x00000C3920B4</td>
</tr>
<tr>
<td>13</td>
<td>Ethernet2/0</td>
<td>172.16..46.1</td>
<td>255.255.255.0</td>
<td>172.16..46.0</td>
<td>0x00000C3920AC</td>
</tr>
<tr>
<td>16</td>
<td>Ethernet2/3</td>
<td>172.16.49.1</td>
<td>255.255.255.0</td>
<td>172.16.49.0</td>
<td>0x00000C3920AF</td>
</tr>
<tr>
<td>17</td>
<td>Ethernet2/4</td>
<td>172.16.52.1</td>
<td>255.255.255.0</td>
<td>172.16.52.0</td>
<td>0x00000C3920B0</td>
</tr>
<tr>
<td>9</td>
<td>Ethernet1/2</td>
<td>172.16.55.1</td>
<td>255.255.255.0</td>
<td>172.16.55.0</td>
<td>0x00000C3920A6</td>
</tr>
<tr>
<td>2</td>
<td>Ethernet 0/1</td>
<td>172.16.56.1</td>
<td>255.255.255.0</td>
<td>172.16.56.0</td>
<td>0x00000C39209D</td>
</tr>
<tr>
<td>15</td>
<td>Ethernet2/2</td>
<td>172.16.57.1</td>
<td>255.255.255.0</td>
<td>172.16.57.0</td>
<td>0x00000C3920AE</td>
</tr>
<tr>
<td>8</td>
<td>Ethernet1/1</td>
<td>172.16.58.1</td>
<td>255.255.255.0</td>
<td>172.16.58.0</td>
<td>0x00000C3920A5</td>
</tr>
<tr>
<td>14</td>
<td>Ethernet2/1</td>
<td>172.16.60.1</td>
<td>255.255.255.0</td>
<td>172.16.60.0</td>
<td>0x00000C3920AD</td>
</tr>
</tbody>
</table>

### Notes
- The above example (Figure 4.3) uses part of the IP MIB discussed for SEQUENCE OF construct.
Encoding

• Basic Encoding Rules (BER)
  • Tag, Length, and Value (TLV)

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class (7-8th bits)</td>
<td>P/C (6th bit)</td>
<td>Tag Number (1-5th bits)</td>
</tr>
</tbody>
</table>

• SNMP Data Types and Tags

<table>
<thead>
<tr>
<th>Type</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJECT IDENTIFIER</td>
<td>UNIVERSAL 6</td>
</tr>
<tr>
<td>SEQUENCE</td>
<td>UNIVERSAL 16</td>
</tr>
<tr>
<td>IpAddress</td>
<td>APPLICATION 0</td>
</tr>
<tr>
<td>Counter</td>
<td>APPLICATION 1</td>
</tr>
<tr>
<td>Gauge</td>
<td>APPLICATION 2</td>
</tr>
<tr>
<td>TimeTicks</td>
<td>APPLICATION 3</td>
</tr>
<tr>
<td>Opaque</td>
<td>APPLICATION 4</td>
</tr>
</tbody>
</table>
Managed Object: Structure

OBJECT:
   sysDescr: { system 1 }
   Syntax: OCTET STRING
   Definition: "A textual description of the entity. This value should include the full name and version identification of the system's hardware type, software operating-system, and networking software. It is mandatory that this only contain printable ASCII characters."
   Access: read-only
   Status: mandatory

Figure 4.17 Specifications for System Description
**Managed Object: Macro**

**OBJECT-TYPE MACRO ::=**
BEGIN
  TYPE NOTATION ::= "SYNTAX" type(TYPE ObjectSyntax)
  "ACCESS" Access
  "STATUS" Status
  VALUE NOTATION ::= value(VALUE ObjectName)

  Access ::= "read-only" | "read-write" | "write-only" | "not-accessible"
  Status ::= "mandatory" | "optional" | "obsolete"

END

*Figure 4.18(a)  OBJECT-TYPE Macro [RFC 1155]*

**sysDescr OBJECT-TYPE**
  SYNTAX DisplayString (SIZE (0..255))
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "A textual description of the entity. This value should include the full name and version identification of the system’s hardware type, software operating-system, and networking software. It is mandatory that this only contain printable ASCII characters."

  ::= {system 1 }

*Figure 4.18(b)  Scalar or Single Instance Macro: sysDescr  [RFC 1213]*
Managed Object: Macro

OBJECT 7

ipAddrTable {ip 20}
Syntax SEQUENCE OF IpAddrEntry
Definition “The table of addressing information relevant to this entity’s IP addresses.”
Access not-accessible
Status : mandatory
Aggregate Object

- A group of objects
- Also called tabular objects
- Can be represented by a table with
  - Columns of objects
  - Rows of instances

Table of Objects

List of Objects

Objects

Notes

- Example: IP address table
- Consists of objects:
  - IP address
  - Interface
  - Subnet mask (which subnet this address belongs to)
  - Broadcast address (value of l.s.b. in IP broadcast address)
  - Largest IP datagram that can be assembled
- Multiple instances of these objects associated with the node
Aggregate Object

OBJECT 7
ipAddrTable {ip 20}
Syntax SEQUENCE OF IpAddrEntry
Definition “The table of addressing information relevant to this entity’s IP addresses.”
Access not-accessible
Status: mandatory
Aggregate Object

OBJECT 1 {ipAdEntAddr} = {internet “123.45.2.1”}
OBJECT 2 {ipAdEntIfIndex} = {“1”}
OBJECT 3 {ipAdEntNetMask} = {internet “255.255.255.0”}
OBJECT 4 {ipAdEntBcastAddr} = {“0”}
OBJECT 5 {ipAdEntReasmMaxSize} = {“12000”}
Aggregate M.O. Macro:  
Table Object

ipAddrTable OBJECT-TYPE
SYNTAX  SEQUENCE OF IpAddrEntry
ACCESS  not-accessible
STATUS  mandatory
DESCRIPTION
  "The table of addressing
  information relevant to this entity's IP
  addresses."
 ::= {ip 20}

ipAddrTable OBJECT-TYPE
 ::= {ip 20}
ipAddrEntry OBJECT-TYPE
 ::= {ipAddrTable 1}
Aggregate M.O. Macro: Entry Object

ipAddrEntry OBJECT-TYPE
SYNTAX  IpAddrEntry
ACCESS  not-accessible
STATUS  mandatory
DESCRIPTION
"The addressing information for one of this entity's IP addresses."

INDEX  { ipAdEntAddr }
::= { ipAddrTable 1 }

IpAddrEntry ::= 
  SEQUENCE  {
    ipAdEntAddr
      IpAddress,
    ipAdEntIfIndex
      INTEGER,
    ipAdEntNetMask
      IpAddress,
    ipAdEntBcastAddr
      INTEGER,
    ipAdEntReasmMaxSize
      INTEGER (0..65535)
  }

Notes
• Index *ipAdEntAddr* uniquely identifies an instance
• May require more than one object in the instance to uniquely identify it
Aggregate M.O. Macro: Columnar Objects

ipAdEntAddr OBJECT-TYPE
SYNTAX IpAddress
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The IP address to which this entry's addressing information pertains."
::= { ipAddrEntry 1 }

ipAdEntReasmMaxSize OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The size of the largest IP datagram which this entity can re-assemble from incoming IP fragmented datagrams received on this interface."
::= { ipAddrEntry 5 }

Notes
Notes

• The objects $TABLE \ T$ and $ENTRY \ E$ are objects that are logical objects. They define the grouping and are not accessible.
• Columnar objects are objects that represent the attributes and hence are accessible.
• Each instance of $E$ is a row of columnar objects 1 through 5.
• Multiple instances of $E$ are represented by multiple rows.
Tabular Representation of Aggregate Object (cont.)

```
<table>
<thead>
<tr>
<th>T.E.1.1</th>
<th>T.E.2.1</th>
<th>T.E.3.1</th>
<th>T.E.4.1</th>
<th>T.E.5.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>T.E.1.2</td>
<td>T.E.2.2</td>
<td>T.E.3.2</td>
<td>T.E.4.2</td>
<td>T.E.5.2</td>
</tr>
<tr>
<td>T.E.1.3</td>
<td>T.E.2.3</td>
<td>T.E.3.3</td>
<td>T.E.4.3</td>
<td>T.E.5.3</td>
</tr>
<tr>
<td>T.E.1.4</td>
<td>T.E.2.4</td>
<td>T.E.3.4</td>
<td>T.E.4.4</td>
<td>T.E.5.4</td>
</tr>
</tbody>
</table>
```

Figure 4.22(b) Example of 5 Columnar Object with 4 Instances (rows)

**Notes**

- Notice that the column-row numeric designation is reverse of what we are used to as row-column
Multiple Instances of Aggregate Managed Object

```
ipAddrTable {1.3.6.1.2.1.4.20}
  ipAddrEntry (1)
    ipAdEntAddr (1)
    ipAdEntIfIndex (2)
    ipAdEntNetMask (3)
    ipAdEntBcastAddr (4)
    ipAdEntReasmMaxSize (5)

Columnar object ID of ipAdEntBcastAddr is (1.3.6.1.2.1.4.20.1.4):
```

```
    iso org dod internet mgmt mib ip ipAddrTable ipAddrEntry ipAdEntBcastAddr
    1 3 6 1 2 1 4 20 1 4

Figure 4.23(a) Columnar objects under ipAddrEntry
```

```
<table>
<thead>
<tr>
<th>Row</th>
<th>ipAdEntAddr</th>
<th>ipAdEntIfIndex</th>
<th>ipAdEntNetMask</th>
<th>ipAdEntBcastAddr</th>
<th>ipAdEntReasmMaxSize</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>123.45.2.1</td>
<td>1</td>
<td>255.255.255.0</td>
<td>0</td>
<td>12000</td>
</tr>
<tr>
<td>2</td>
<td>123.45.3.4</td>
<td>3</td>
<td>255.255.0.0</td>
<td>1</td>
<td>12000</td>
</tr>
<tr>
<td>3</td>
<td>165.8.9.25</td>
<td>2</td>
<td>255.255.255.0</td>
<td>0</td>
<td>10000</td>
</tr>
<tr>
<td>4</td>
<td>9.96.8.138</td>
<td>4</td>
<td>255.255.255.0</td>
<td>0</td>
<td>15000</td>
</tr>
</tbody>
</table>

Figure 4.23(b) Object instances of ipAddrTable (1.3.6.1.2.1.4.20)

```
<table>
<thead>
<tr>
<th>Columnar Object</th>
<th>Row # in (b)</th>
<th>Object Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipAdEntAddr</td>
<td>2</td>
<td>{1.3.6.1.2.1.4.20.1.1.123.45.3.4}</td>
</tr>
<tr>
<td>ipAdEntIfIndex</td>
<td>3</td>
<td>{1.3.6.1.2.1.4.20.1.2.165.8.9.25}</td>
</tr>
<tr>
<td>ipAdEntBcastAddr</td>
<td>1</td>
<td>{1.3.6.1.2.1.4.20.1.4.123.45.2.1}</td>
</tr>
<tr>
<td>ipAdEntReasmMaxSize</td>
<td>4</td>
<td>{1.3.6.1.2.1.4.20.1.5.9.96.8.138}</td>
</tr>
</tbody>
</table>

Figure 4.23(c) Object Id for specific instance
SMI Definition STD 16 / 1155 RFC

RFC1155-SMI DEFINITIONS ::= BEGIN

  EXPORTS -- EVERYTHING
  internet, directory, mgmt, experimental, private, enterprises,
  OBJECT-TYPE, ObjectName, ObjectSyntax, SimpleSyntax,
  ApplicationSyntax, NetworkAddress, IpAddress, Counter, Gauge,
  TimeTicks, Opaque;

  -- the path to the root

  internet   OBJECT IDENTIFIER ::= { iso org(3) dod(6) 1 }
  directory  OBJECT IDENTIFIER ::= { internet 1 }
  mgmt       OBJECT IDENTIFIER ::= { internet 2 }
  experimental OBJECT IDENTIFIER ::= { internet 3 }
  private    OBJECT IDENTIFIER ::= { internet 4 }
  enterprises OBJECT IDENTIFIER ::= { private 1 }

---

Notes

- EXPORTS identifies the objects that any other module could import.
SMI Definition STD 16 / 1155 RFC (cont.)

-- definition of object types

OBJECT-TYPE MACRO ::= BEGIN
    TYPE NOTATION ::= "SYNTAX" type (TYPE ObjectSyntax)
                    "ACCESS" Access
                    "STATUS" Status
    VALUE NOTATION ::= value (VALUE ObjectName)

    Access ::= "read-only" | "read-write" | "write-only" | "not-accessible"
    Status ::= "mandatory" | "optional" | "obsolete"
END
SMI Definition STD 16 / 1155 RFC (cont.)

-- names of objects in the MIB

ObjectName ::= OBJECT IDENTIFIER

-- syntax of objects in the MIB

ObjectSyntax ::= CHOICE {
    simple
        SimpleSyntax,

    application-wide
        ApplicationSyntax
}

Notes
SMI Definition STD 16 / 1155 RFC (cont.)

SimpleSyntax ::=  
    CHOICE {  
        number  
            INTEGER,  
        string  
            OCTET STRING,  
        object  
            OBJECT IDENTIFIER,  
        empty  
            NULL  
    }

-----------------------------------------------

ApplicationSyntax ::=  
    CHOICE {  
        address  
            NetworkAddress,  
        counter  
            Counter,  
        gauge  
            Gauge,  
        ticks  
            TimeTicks,  
        arbitrary  
            Opaque  
    }

    -- other application-wide types, as they are defined,  
    will be added here  
}
SMI Definition STD 16 / 1155 RFC (cont.)

-- application-wide types

NetworkAddress ::= CHOICE {
    internet
    IpAddress
}
IpAddress ::= [APPLICATION 0]          -- in network-byte order
    IMPLICIT OCTET STRING (SIZE (4))
Counter ::= [APPLICATION 1]
    IMPLICIT INTEGER (0..4294967295)
Gauge ::= [APPLICATION 2]
    IMPLICIT INTEGER (0..4294967295)
TimeTicks ::= [APPLICATION 3]
    IMPLICIT INTEGER (0..4294967295)
Opaque ::= [APPLICATION 4]          -- arbitrary ASN.1 value,
    IMPLICIT OCTET STRING   -- "double-wrapped"

END

Notes
MIB

- MIB-II (RFC 1213) is superset of MIB-I.
- Objects that are related grouped into object groups.
- MIB module comprises module name, imports from other modules, and definitions of current module.
- RFC 1213 defines eleven groups; expanded later.

Figure 4.26 Internet MIB-II Group

Notes
System Group

![Diagram of System Group]

Figure 4.27 System Group

<table>
<thead>
<tr>
<th>Entity</th>
<th>OID</th>
<th>Description (brief)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysDescr</td>
<td>system 1</td>
<td>Textual description</td>
</tr>
<tr>
<td>sysObjectID</td>
<td>system 2</td>
<td>OBJECT IDENTIFIER of the entity</td>
</tr>
<tr>
<td>sysUpTime</td>
<td>system 3</td>
<td>Time (in hundredths of a second since last reset)</td>
</tr>
<tr>
<td>sysContact</td>
<td>system 4</td>
<td>Contact person for the node</td>
</tr>
<tr>
<td>sysName</td>
<td>system 5</td>
<td>Administrative name of the system</td>
</tr>
<tr>
<td>sysLocation</td>
<td>system 6</td>
<td>Physical location of the node</td>
</tr>
<tr>
<td>sysServices</td>
<td>system 7</td>
<td>Value designating the layer services provided by the entity</td>
</tr>
</tbody>
</table>

Notes
sysServices

sysServices OBJECT-TYPE
   SYNTAX  INTEGER (0..127)
   ACCESS  read-only
   STATUS  mandatory
   DESCRIPTION
      "A value which indicates the set of services that this entity primarily offers.

The value is a sum. This sum initially takes the value zero, then, for each layer, L, in the range 1 through 7, that this node performs transactions for, 2 raised to (L - 1) is added to the sum. For example, a node which performs primarily routing functions would have a value of 4 (2^(3-1)). In contrast, a node which is a host offering application services would have a value of 72 (2^(4-1) + 2^(7-1)). Note that in the context of the Internet suite of protocols, values should be calculated accordingly:

layer functionality
   1  physical (e.g., repeaters)
   2  datalink/subnetwork (e.g., bridges)
   3  internet (e.g., IP gateways)
   4  end-to-end (e.g., IP hosts)
   7  applications (e.g., mail relays)

For systems including OSI protocols, layers 5 and 6 may also be counted."
::= { system 7 }

Notes
Interfaces Group

Legend: INDEX in bold

Figure 4.28 Interfaces Group

Notes
Notes
- Interfaces MIB limited by maximum number of physical ports
- A physical port may have several conceptual ports e.g., channels in cable access network
- ifMIB {mib-2 31} created to extend interfaces MIB
- ifMIB specifies extension in generic manner
- Specific technology related MIBs supplement details on the conceptual ports
- ifIndex in interfaces MIB can exceed the maximum number of physical ports
- ifStack definition accommodates interface sublayers
Interface Sublayers

(a) Interface Stacked layers

(b) Interface Multiplexed layers

Figure 4.29 Interface Sublayers
ifEntry

IfEntry OBJECT-TYPE
SYNTAX IfEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"An interface entry containing objects at the subnetwork layer and below for a particular interface."
INDEX {ifIndex}
::= {ifTable 1}

Notes

• ifEntry specifies the objects in a row in the ifTable.
• Each interface is defined as a row in the table.
### ifType

ifType OBJECT-TYPE
SYNTAX  INTEGER {
  other(1), -- none of the following
  regular1822(2),
  hdh1822(3),
  ddn-x25(4),
  rfc877-x25(5),
  ethernet-csmacd(6),
  iso88023-csmacd(7),
  iso88024-tokenBus(8),
  iso88025-tokenRing(9),
  iso88026-man(10),
  starLan(11),
  proteon-10Mbit(12),
  proteon-80Mbit(13),
  hyperchannel(14),
  fddi(15),
  lapb(16),
  sdlc(17),
  ds1(18), -- T-1
  e1(19), -- european equiv. of T-1
  basicISDN(20),
  primaryISDN(21), -- proprietary serial
  propPointToPointSerial(22),
  ppp(23),
  ........
}

---

**Notes**

- Type of interface below the network layer defined as enumerated integer.
IP Group

Note

- ipForwarding: Gateway(1) and Router(2)
- IP Address Table contains table of IP addresses
- IP Route Table contains an entry for each route
- IP Network-to-Media Table is address translation table mapping IP addresses to physical addresses
### IP Address Table

![Diagram of IP Address Table]

<table>
<thead>
<tr>
<th>Entity</th>
<th>OID</th>
<th>Description (brief)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipAddrTable</td>
<td>ip 20</td>
<td>Table of IP addresses</td>
</tr>
<tr>
<td>ipAddrEntry</td>
<td>IpAddrTable 1</td>
<td>One of the entries in the IP address table</td>
</tr>
<tr>
<td>ipAdEntAddr</td>
<td>IpAddrEntry 1</td>
<td>The IP address to which this entry's addressing information pertains</td>
</tr>
<tr>
<td>ipAdEntIfIndex</td>
<td>IpAddrEntry 2</td>
<td>Index value of the entry, same as ifIndex</td>
</tr>
<tr>
<td>ipAdEntNetMask</td>
<td>IpAddrEntry 3</td>
<td>Subnet mask for the IP address of the entry</td>
</tr>
<tr>
<td>ipAdEntBcastAddr</td>
<td>IpAddrEntry 4</td>
<td>Broadcast address indicator bit</td>
</tr>
<tr>
<td>ipAdEntReasmMaxSize</td>
<td>IpAddrEntry 5</td>
<td>Largest IP datagram that can be reassembled on this interface</td>
</tr>
</tbody>
</table>

**Legend:** INDEX in bold

**Figure 4.30 IP Address Table**

### Notes

<table>
<thead>
<tr>
<th>Entity</th>
<th>OID</th>
<th>Description (brief)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipAddrTable</td>
<td>ip 20</td>
<td>Table of IP addresses</td>
</tr>
<tr>
<td>ipAddrEntry</td>
<td>IpAddrTable 1</td>
<td>One of the entries in the IP address table</td>
</tr>
<tr>
<td>ipAdEntAddr</td>
<td>IpAddrEntry 1</td>
<td>The IP address to which this entry's addressing information pertains</td>
</tr>
<tr>
<td>ipAdEntIfIndex</td>
<td>IpAddrEntry 2</td>
<td>Index value of the entry, same as ifIndex</td>
</tr>
<tr>
<td>ipAdEntNetMask</td>
<td>IpAddrEntry 3</td>
<td>Subnet mask for the IP address of the entry</td>
</tr>
<tr>
<td>ipAdEntBcastAddr</td>
<td>IpAddrEntry 4</td>
<td>Broadcast address indicator bit</td>
</tr>
<tr>
<td>ipAdEntReasmMaxSize</td>
<td>IpAddrEntry 5</td>
<td>Largest IP datagram that can be reassembled on this interface</td>
</tr>
</tbody>
</table>
### IP Routing Table

**Figure 4.31 IP Routing Table**

<table>
<thead>
<tr>
<th>Entity</th>
<th>OID</th>
<th>Description (brief)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipRouteTable</td>
<td>ip 21</td>
<td>IP routing table</td>
</tr>
<tr>
<td>ipRouteEntry</td>
<td>ipRouteTable 1</td>
<td>Route to a particular destination</td>
</tr>
<tr>
<td>ipRouteDest</td>
<td>ipRouteEntry 1</td>
<td>Destination IP address of this route</td>
</tr>
<tr>
<td>ipRouteIfIndex</td>
<td>ipRouteEntry 2</td>
<td>Index of interface, same as iFIndex</td>
</tr>
<tr>
<td>ipRouteMetric1</td>
<td>ipRouteEntry 3</td>
<td>Primary routing metric for this route</td>
</tr>
<tr>
<td>ipRouteMetric2</td>
<td>ipRouteEntry 4</td>
<td>An alternative routing metric for this route</td>
</tr>
<tr>
<td>ipRouteMetric3</td>
<td>ipRouteEntry 5</td>
<td>An alternative routing metric for this route</td>
</tr>
<tr>
<td>ipRouteMetric4</td>
<td>ipRouteEntry 6</td>
<td>An alternative routing metric for this route</td>
</tr>
<tr>
<td>ipRouteNextHop</td>
<td>ipRouteEntry 7</td>
<td>IP address of the next hop</td>
</tr>
<tr>
<td>ipRouteType</td>
<td>ipRouteEntry 8</td>
<td>Type of route</td>
</tr>
<tr>
<td>ipRouteProto</td>
<td>ipRouteEntry 9</td>
<td>Routing mechanism by which this route was learned</td>
</tr>
<tr>
<td>ipRouteAge</td>
<td>ipRouteEntry 10</td>
<td>Number of seconds since routing was last updated</td>
</tr>
<tr>
<td>ipRouteMask</td>
<td>ipRouteEntry 11</td>
<td>Mask to be logically ANDed with the destination address before comparing with the ipRouteDest field</td>
</tr>
<tr>
<td>ipRouteMetric5</td>
<td>ipRouteEntry 12</td>
<td>An alternative metric for this route</td>
</tr>
<tr>
<td>ipRouteInfo</td>
<td>ipRouteEntry 13</td>
<td>Reference to MIB definition specific to the routing protocol</td>
</tr>
</tbody>
</table>
IP Address Translation Table

Figure 4.32  IP Address Translation Table

<table>
<thead>
<tr>
<th>Entity</th>
<th>OID</th>
<th>Description (brief)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipNetToMediaTable</td>
<td>ip 22</td>
<td>Table mapping IP addresses to physical addresses</td>
</tr>
<tr>
<td>ipNetToMediaEntry</td>
<td>IpNetToMediaTable 1</td>
<td>IP address to physical address for the particular interface</td>
</tr>
<tr>
<td>ipNetToMediaIfIndex</td>
<td>IpNetToMediaEntry 1</td>
<td>Interfaces on which this entry’s equivalence is effective; same as ifIndex</td>
</tr>
<tr>
<td>ipNetToMediaPhysAddress</td>
<td>IpNetToMediaEntry 2</td>
<td>Media dependent physical address</td>
</tr>
<tr>
<td>ipNetToMediaNetAddress</td>
<td>IpNetToMediaEntry 3</td>
<td>IP address</td>
</tr>
<tr>
<td>ipNetToMediaType</td>
<td>IpNetToMediaEntry 4</td>
<td>Type of mapping</td>
</tr>
</tbody>
</table>
ICMP Group

Figure 4.34  ICMP Group

Notes

• Objects associated with ping
  • icmpOutEchos # ICMP echo messages sent
  • icmpInEchoReps # ICMP echo reply messages received

• Objects associated with traceroute/tracert
  • icmpInTimeExcs # ICMP time exceeded messages received
Notes

- Connection-oriented transport protocol group
- Has one table
TCP Connection Table

Figure 4.36  TCP Connection Table

Notes

<table>
<thead>
<tr>
<th>Entity</th>
<th>OID</th>
<th>Description (brief)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcpConnTable</td>
<td>tcp 13</td>
<td>TCO connection table</td>
</tr>
<tr>
<td>tcpconnEntry</td>
<td>TcpConnTable 1</td>
<td>Information about a particular TCP connection</td>
</tr>
<tr>
<td>tcpConnState</td>
<td>TcpConnEntry 1</td>
<td>State of the TCP connection</td>
</tr>
<tr>
<td>tcpConnLocalAddress</td>
<td>TcpConnEntry 2</td>
<td>Local IP address</td>
</tr>
<tr>
<td>tcpConnLocalPort</td>
<td>TcpConnEntry 3</td>
<td>Local port number</td>
</tr>
<tr>
<td>tcpConnRemAddress</td>
<td>TcpConnEntry 4</td>
<td>Remote IP address</td>
</tr>
<tr>
<td>tcpConnRemPort</td>
<td>TcpConnEntry 5</td>
<td>Remote port number</td>
</tr>
</tbody>
</table>
UDP Group

Figure 4.37 UDP Group

Notes

• Connectionless transport protocol group
• Has one table, UDP table

<table>
<thead>
<tr>
<th>Entity</th>
<th>OID</th>
<th>Description (brief)</th>
</tr>
</thead>
<tbody>
<tr>
<td>udpInDatagrams</td>
<td>udp 1</td>
<td>Total number of datagrams delivered to the users</td>
</tr>
<tr>
<td>udpNoPorts</td>
<td>udp 2</td>
<td>Total number of received datagrams for which there is no application</td>
</tr>
<tr>
<td>udpInErrors</td>
<td>udp 3</td>
<td>Number of received datagrams with errors</td>
</tr>
<tr>
<td>udpOutDatagrams</td>
<td>udp 4</td>
<td>Total number of datagrams sent</td>
</tr>
<tr>
<td>udpTable</td>
<td>udp 5</td>
<td>UDP Listener table</td>
</tr>
<tr>
<td>udpEntry</td>
<td>udpTable 1</td>
<td>Information about a particular connection or UDP listener</td>
</tr>
<tr>
<td>udpLocalAddress</td>
<td>udpEntry 1</td>
<td>Local IP address</td>
</tr>
<tr>
<td>udpLocalPort</td>
<td>udpEntry 2</td>
<td>Local UDP port</td>
</tr>
</tbody>
</table>