SNMP

Simple Network Management Protocol

Agenda

- **Network Management Basics**
- **SNMP**
  - Basics
  - SMI
  - MIB
- **RMON**
- **SNMPv2**
- **Product Examples**

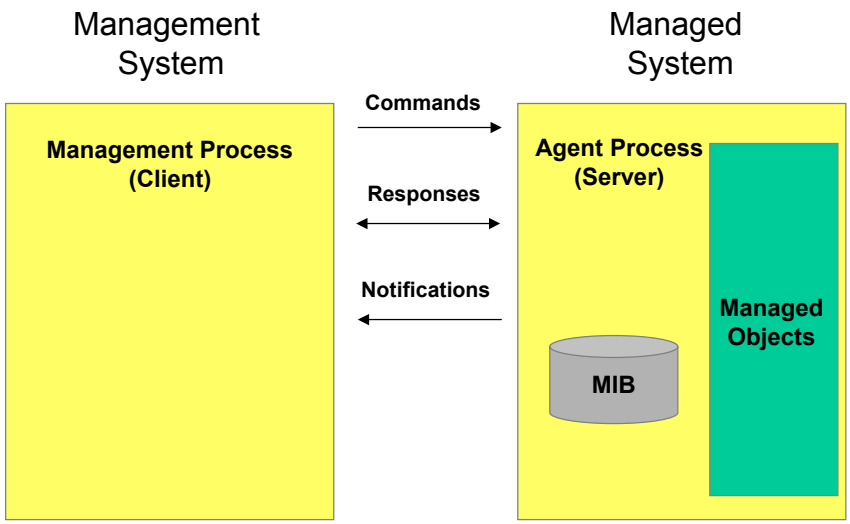
What is it?

- **A network management should...**
  - automate the process of monitoring and adjusting the performance of a network
  - trigger alarms when special events occur
  - not burden the network
- **Two important implementations**
  - IETF Network Management based on SNMP
  - OSI Network Management Architecture
- **SNMP is the most important implementation**
  - Industry standard

Four Basic Elements

- **Management Process (Client)**
  - Central workstation
- **Agent Process (Server)**
  - Routers, hosts, ...
- **Management Information Base (MIB)**
  - The logical structure of this database is defined in the "Structure of Management Information" (SMI)
- **Communication Protocol**
  - Special PDU´s (e.g. SNMP)

Client/Server System



The MIB

- **The whole management information is organized as tree-structure**
  - Branches represent logical categories
  - Leaves contain information about objects
- **Tree can be locally or globally valid**
- **ISO administers one of the branches of the global tree**
  - iso(1)

Requirement for a SMI

- **Structure of Management Information (SMI)**
  - Specifies how information about managed objects is to be represented
- **This information representation is implemented using a restricted version of the ISO's Abstract Syntax Notation One (ASN.1)**
  - ASN.1 was designed to describe data types for the OSI presentation layer
  - Similar to a higher level programming language

ISO/OSI Network Management Model

- **ISO/IEC 7498-4**
  - OSI network management framework
- **ISO/IEC 9595 or CCITT X.710**
  - Common Management Information Service (CMIS)
  - Manages the OSI-MIB
- **ISO/IEC 9596 or CCITT X.711**
  - Common Management Information Protocol (CMIP)
  - Layer 7, connection oriented
- **Note: Also IEEE 802.1B LAN/MAN management standard uses ISO's CMIP**

IETF Management Framework

- **Late 1980s: IAB realized the demand for a general Internet management architecture**
- **Initially three proposals:**
  - High-level entity management system (HEMS)
  - OSI-based system using CMIS and CMIP
  - An extended version of the Simple Gateway Monitoring Protocol (SGMP)
    - Used in the early Arpanet (RFC 1028, historic)
- **IAB decided for a extended version of SGMP called Simple Network Management Protocol (SNMP)**

IETF Management Framework

- **SNMP was intended as short-time solution only (!)**
  - Should be replaced later by an OSI approach
- **So IAB formed another working group for "CMOT"**
  - CMIS/CMIP over TCP/IP = CMOT
  - RFC 1052 reflects these efforts
- **But this development could not keep up with time**
  - Too complex
  - No working stacks at that time
- **Hence SNMP became an industrial standard**

CMOT

- **CMIP over TCP/IP is known as CMOT**
  - RFC-1095
- **In 1990 the IAB recommended CMOT**
  - Although the IETF regards this work as "historic"
  - However, some vendors use CMOT
- **Main differences to SNMP:**
  - TCP instead of UDP
  - Connection-oriented application layer
  - CMOT's application layer is built on OSI services
  - MIB-II-OIM instead of SNMP's MIB-II

CMOT

- **CMOT is much more sophisticated**
  - CMIP object definitions are more comprehensive
- **In CMOT the management application relies on three OSI services:**
  - Common Management Information Service Element (CMISE)
  - Remote Operation Service Element (ROSE)
  - Association Control Service Element (ACSE)
- **And a Lightweight Presentation Protocol (LPP)**
  - As presentation layer

Agenda

- **Network Management Basics**
- **SNMP**
  - Basics
  - SMI
  - MIB
- **RMON**
- **SNMPv2**
- **Product Examples**

SNMP

- **What does SNMP ?**
  - Using SNMP, a NMS can set and retrieve variables of a MIB
- **Why "Simple" Network Management Protocol?**
  - Because the agents require only minimal software
  - Only the network management system (NMS, client side!) must provide enough CPU and memory resources
  - Only a small set of messages
    - Client: Get, GetNext, Set
    - Agents: Response, event notification (trap)

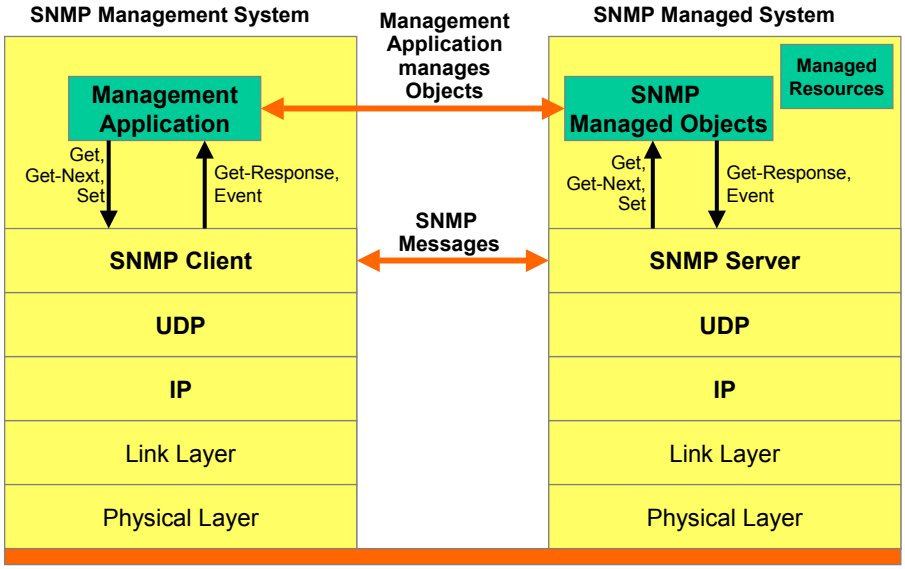
SNMP

- **Connectionless Communication**
  - Via UDP
    - port numbers 161 (Agent) and 162 (Client)
- **SNMP is described in RFC 1157**
  - IAB recommends the usage of the Internet MIB (RFC-1156) and at least one of the recommended management protocols, SNMP or CMOT
- **NMI acts as SNMP client**
  - Polls SNMP Agents periodically and collects performance statistics from them
- **Managed device is called SNMP Agent**
  - Runs SNMP server software
  - Can send so-called TRAPS to the NMI in case of an event (alarm)

Typical Agents

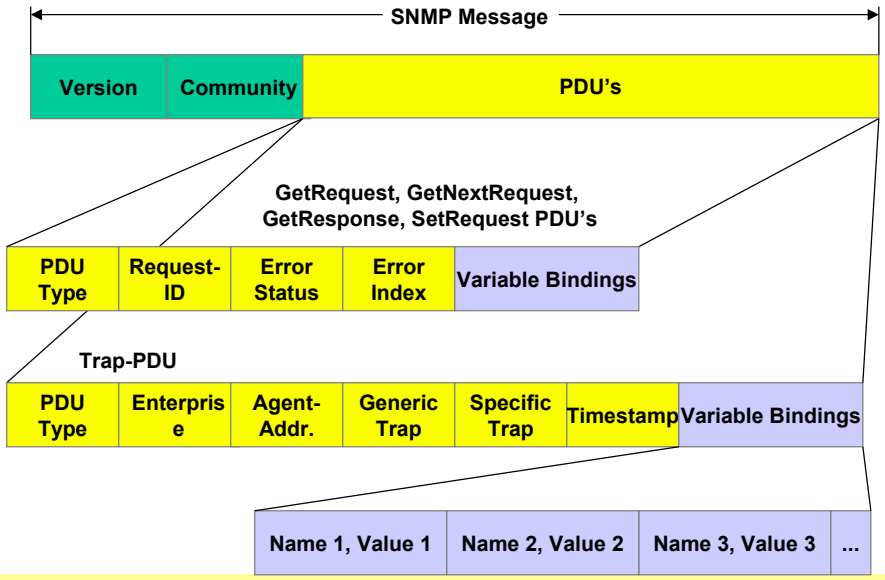
- **Typical SNMP agents:**
  - Wiring hubs
  - Network servers
  - Hosts with network interface cards
  - Internetworking devices, such as bridges and routers
  - Test equipment, such as monitors and analyzers
  - Uninterruptible Power Supplies (UPS)
- **Additionally, SNMP supplemental's are commonly supported, such as**
  - HTTP/HTML based systems
  - Web-based Enterprise Management (WBEM) solution
  - Java Management Application Programming Interface (JMAPI) proposed by SunSoft

SNMP Architecture



 SNMP, v4.0 17

SNMP Message Format



 SNMP, v4.0 18

SNMP Header Entries

- **Version**
  - Specifies the protocol version (0..SNMPv1)
- **Community**
  - Contains a "community" password to protect the session
  - Note: cleartext!
    - Security hole if set commands are allowed
    - Hence many vendors disallow set Commands or provide proprietary cryptographic solutions
    - Solved in SNMPv2
- **Request-ID**
  - Identifies corresponding requests/responds

SNMP Header Entries

- **Error Status**
  - 0 (No Error), 1 (Too Big), 2 (No Such Name), 3 (Bad value), 4 (Read Only), 5 (General Error)
  - In Requests always set to zero
- **Error Index**
  - Points to the variable of the Request message that caused the error
  - In Requests always set to zero
- **Variable Bindings**
  - List of object names plus associated values
  - In Requests the values are always set to zero
- **Enterprise**
  - Type of management agent who generated the trap
  - Based on sysObjectID

SNMP Header Entries

- **Agent-Addr**
  - Address of management agent
- **Generic Trap**
  - 0 (Cold Start), 1 (Warm Start), 2 (Link Down), 3 (Link Up), 4 (Authentication Failure), 5 (EGP Neighbor Loss), 6 (Enterprise Specific)
- **Specific Trap**
  - Special vendor specific trap
- **Time-Stamp**
  - Contains system time at which the trap occured
  - Based on sysUpTime

Agenda

- **Network Management Basics**
- **SNMP**
  - Basics
  - SMI
  - MIB
- **RMON**
- **SNMPv2**
- **Product Examples**

SMI

- **Structure of Management Information (SMI)**
  - RFC 1155, RFC 1212, RFC 1215
- **SMI organizes, describes, and names information**
  - Thus providing accessibility
- **Each managed object must consist of**
  - a name
  - a syntax
  - and an encoding
- **A MIB contains these objects**

Managed Objects

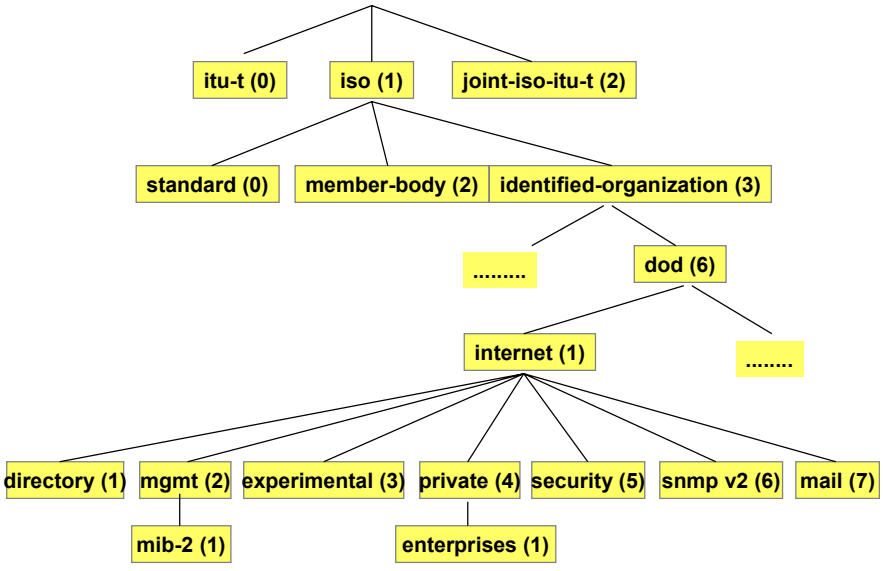
- **Name**
  - Is also called Object Identifier (OID)
  - Uniquely identifies the object
- **Syntax**
  - Defines the data type (integer, string of bytes, etc.)
  - Described with the Abstract Syntax Notation One (ASN.1)
- **Encoding**
  - Describes how the information is transmitted between machines
  - Using so-called Basic Encoding Rules (BER) to specify the transfer syntax

SMI – Names

- **Each object must have a name**
  - An object is either a device
  - or a characteristic of a device
- **Name = sequence of integers separated by dots**
  - Also known as Object Identifier
  - Represents the tree structure of a MIB
- **Internet sub-tree uses prefix 1.3.6.1**
  - Administered by IANA
  - RFC 1700

 SNMP, v4.0 25

SNMP Related OID Subtree

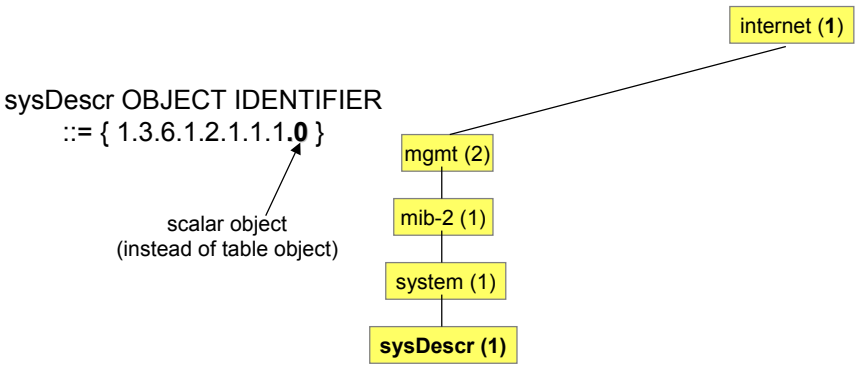


 SNMP, v4.0 26

Name Example

ASN.1 Notation:

```
mgmt OBJECT IDENTIFIER ::= { internet 2 }
mib-2 OBJECT IDENTIFIER ::= { mgmt 1 }
system OBJECT IDENTIFIER ::= { mib-2 1 }
sysDescr OBJECT IDENTIFIER ::= { system 1 }
```



MIB Internet Branches

- **directory (1)**
  - was reserved for the OSI directory within the Internet
- **mgmt (2)**
  - contains all Internet Standard MIBs
- **experimental (3)**
  - is used for Internet experiments (IANA)
  - if certain objects are approved they move to an official branch
- **private (4)**
  - is used by vendors to register their own MIBs

SMI – Syntax

- **According ISO/OSI the ASN.1 is a presentation layer function**
  - For simplicity, SNMP uses only a subset of ASN.1
- **ASN.1 defines structured information in a machine-independent fashion**
  - Provides basic data types
  - All data types can be defined with this basic data types
- **ASN.1 describes:**
  - Types and values
  - Macros
  - Modules

ASN.1 – Types

- **Primitive data types (aka Simple Types)**
  - INTEGER, OCTET STRING, OBJECT IDENTIFIER, NULL
- **Subtypes**
  - E.g. INTEGER (0..255)
  - E.g. special cases of OCTET STRING
    - DisplayString: only printable ASCII characters
    - PhysAddress: for MAC addresses
    - octetBitstring: for bitstrings longer than 32 bits
- **Constructor types**
  - SEQUENCE, SEQUENCE OF
  - Define tables and rows within those tables

ASN.1 – Types

- **Defined types**
  - RFC 1213: NetworkAddress, IpAddress, Counter, Gauge, TimeTicks, and Opaque
- **Tagged types**
  - To differentiate between defined types
  - Types are defined using other defined types as basis and by adding additional information (tags)
  - Tags consist of class and number for the contained type
    - class universal: e.g. INTEGER, OCTET-STRING, ...
    - class application: e.g. IPAddress, Counter, ...
    - class context-specific: SNMP PDUs such as GetRequest, GetResponse, ...
    - class private: for enterprise specific applications

ASN.1 – Macros

- **Macros extend the ASN.1 language**
  - They are used to describe objects
- **Example for object "tcpInSegs"**

```
tcpInSegs OBJECT-TYPE
      SYNTAX  Counter
      ACCESS  read-only
      STATUS  mandatory
      ::= { tcp 10 }
```

  - Used Macro: OBJECT-TYPE
    - Is the most important macro (RFC 1213)
  - The object "tcpInSegs" represents a counter
  - This object can be accessed read-only under the name "tcp 10"
    - That is, tcpInSegs is the 10th object in the tcp group
  - This object is mandatory for all devices in mib-2.tcp

ASN.1 – Modules

- **Modules represent a collection of types, values and macros**
- **Structure of a module**
  - Module name
    - e.g. RMON-MIB
  - Body is enclosed with BEGIN and END
  - IMPORT-statement is used to collect types, values and macros from other modules
    - e.g. the type "Counter" from the module RFC1155-SMI
    - e.g. the macro "OBJECT-TYPE" from the module RFC-1212

Example Module

- **For example the RMON MIB module contains the following code:**

```
RMON-MIB DEFINITIONS ::= BEGIN
        IMPORTS
           Counter                 FROM RFC1155-SMI
           DisplayString           FROM RFC1158-MIB
           mib-2                   FROM RFC1213-MIB
           OBJECT-TYPE             FROM RFC-1212
           TRAP-TYPE               FROM RFC-1215;
    -- This is a comment line
    -- Remote Network Monitoring MIB
         rmon    OBJECT IDENTIFIER ::= { mib-2 16 }
         ...
END
```

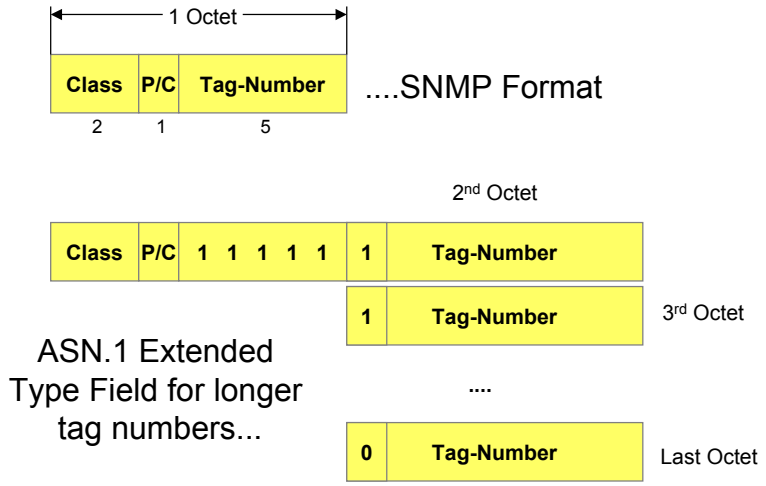

SMI – Encoding

- **Basic Encoding Rules (BER)**
  - Defines how to convert ASN.1 data types into bit patterns
  - Specified in ISO 8825-1
- **BER uses "Type-Length-Value" (TLV) coding**
  - Also other names used, such as
    - "Identifier-Length-Contents"
    - "Tag-Length-Value"
  - Each message is structured accordingly

BER – Type Field

- **Type field**
  - Contains the ASN.1 tag (class and number) to identify the data in the Value field
- **SNMP uses only one octet for the Type field**
  - Bits 1-5: tag numbers (0-30)
    - 2 .. Integer, 4 .. Octetstring, 5 ..Null, 6 .. Object Identifier, 16 .. Sequence
  - Bit 6: P/C bit (Primitive/Constructed)
  - Bits 7,8: class
    - 00..Universal, 01..Application, 10..Context Specific, 11..Private

Type Field



BER – Length and Value Fields

- **Length field**
  - Specifies the number of bytes in the Value field
  - Size of length field: 1-127
    - MSB = 0 means: 1 octet
    - MSB = 1 means: first octet specifies the number of octets of the length field
- **Value field**
  - 0 - (2^1008-1) octets theoretically possible
  - Contains either an
    - Integer
    - ASCII character
    - OBJECT-IDENTIFIER

Value – OBJECT-IDENTIFIER

- **OBJECT-IDENTIFIERs**
  - Are also encoded
  - Encoding trick for first (X) and second (Y) subidentifer
    - Encoded value = (40 * X) + Y
    - To save one octet
- **Example:**
  - { iso org(3) dod(6) internet(1) mgmt(2) mib-2 (1) 1 }
  - encoded as 0x 2B 06 01 02 01 01

Example BER for a SNMP Message

GetRequest für sysDescr (1.3.6.1.2.1.1.1)

```
30      29      02      01      00
sequence, len=41, integer, len=1, version=0
04      06      70      75      62      6C      69      63
string, len=6, communityname= PUBLIC
A0      1C      02      04      05      AE      56      02
GetReq., len=28, integer, len=4, request-id=05AE5602
02      01      00      02      01      00
integer, len=1, status=0, integer, len=1, error-index=0
30      0E      30      0C      06      08
sequence, len=14, sequence, len=12, object-id, len=8
2B      06      01      02      01      01      01      00
1.3 .   6   .   1   .   2   .   1   .   1   .   1   .   0
05      00
null, len=0
```

Agenda

- **Network Management Basics**
- **SNMP**
  - Basics
  - SMI
  - MIB
- **RMON**
- **SNMPv2**
- **Product Examples**

MIB

- **All Internet Standard MIBs are in the mgmt (2) sub-tree**
- **MIB-I: 8 objects**
  - System, Interfaces, Address Translation, IP, ICMP, TCP, UDP, and EGP
  - RFC 1156, May 1990
- **RFC 1212: Concise MIB**
  - Reason: several proprietary MIB's occur
  - Demand for consistent format for MIB modules
- **MIB-II: Replaced MIB-I**
  - RFC 1213, March 1991

MIB Entries

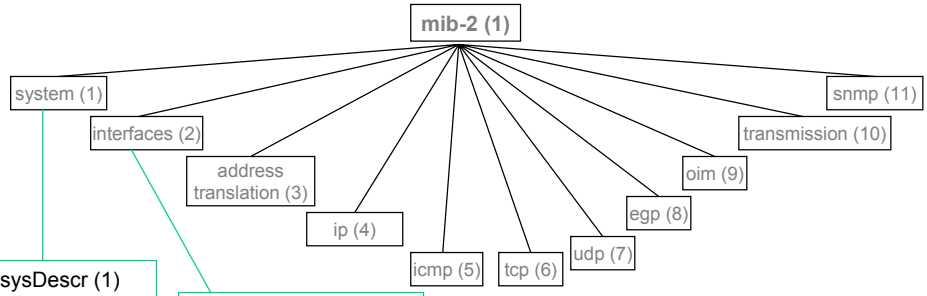
- **MIB objects**
  - Are described with the ASN.1 OBJECT-TYPE macro
- **MIB table entries**
  - Are also described using the OBJECT-TYPE macro
  - plus the SEQUENCE OF constructor type to define and attach the table at the tree
  - plus the INDEX statement to define the row
  - plus the SEQUENCE constructor type to define the columns of a row

Example: ASN.1 Table Objects

```
udpTable  OBJECT-TYPE
  SYNTAX SEQUENCE OF UdpEntry
  ACCESS not-accessible
  STATUS mandatory
  DESCRIPTION
   "A table containing UDP listener
   information.
::= { udp 5 }

udpEntry      OBJECT-TYPE
   SYNTAX  UdpEntry
   ACCESS  not-accessible
   STATUS  mandatory
   DESCRIPTION
      "Information about a particular
      current UDP listener."
   INDEX { udpLocalAddress, udpLocalPort }
::= { udpTable 1 }

UdpEntry ::=
      SEQUENCE {
          udpLocalAddress IpAddress,
          udpLocalPort INTEGER (0..65535)
      }

udpLocalAddress  OBJECT-TYPE
          SYNTAX    IpAddress
          ACCESS    read-only
          STATUS    mandatory
          DESCRIPTION
 ::= { udpEntry 1 }

udpLocalPort    OBJECT-TYPE
    SYNTAX   INTEGER (0..65535)
    ACCESS   read-only
    STATUS   mandatory
    DESCRIPTION
      "The local port number
      for this UDP listener."
  ::= { udpEntry 2 }
```

Note that the sequence name is the same as the row name except that it begins with an uppercase letter (Convention)

MIB-II Excerpt



sysDescr (1)
sysObjID (2)
sysUpTime (3)
sysContact (4)
sysName (5)
sysLocation (6)
sysServices (7)

ifNumber (1)
ifTable (2)
  ifEntry (1)
    ifIndex (1)
    ifDescr (2)
    ifType (3)
    ifMtu (4)
    ifSpeed (5)
    ifPhysicalAddress (6)
    ifAdminStatus (7)
    ..... (22)

- ifNumber (1) = number of interfaces on the device
- ifTable (2) contains all interface information
- Each interface represents a row entry in the table
- Each row entry has 22 columns

Variables of the "System" Group

- sysDescr (1): Display String, RO, M
  - Description of the hardware, operating system, etc
- sysObjID (2): Object Identifier, RO, M
  - Number-string representing the OID of the manufacturer's private MIB for this device
- sysUpTime (3): TimeTicks, RO, M
  - This Agent's last initialization time
- sysContact (4), sysName (5), sysLocation (6): Display Strings, RW, M
  - Can be utilized by the network manager
- sysServices (7): Integer, RO, M
  - A number specifying delivered services/layer

Variables of the "Interface" Group

- ifNumber (1): Integer, RO, M
  - Number of network interfaces
- ifTable (2): Sequence of ifEntry, NA, M
  - List of ifEntry (2.1) using the following entries:
- ifDescre (2.1.2): Display String RO, M
  - Information from manufacturer
- ifType (2.1.3): Integer, RO, M
  - 05 ... X.25 (RFC 877), 06 ... Ethernet, 07 ... 802.3,
  - 09 ... 802.5, 10 ... 802.6 (MAN), 15 ... FDDI,16 ... LAPD,
  - 17 ... Sdlc, 19 ... ES1, 20 ... Basic ISDN, 21 ... Primary ISDN,
  - 23 ... PPP, 28 ... Slip, 32 ... Frame Relay

Variables of the "Interface" Group

- ifMtu (2.1.4): Integer, RO, M
- ifPhysAddress (2.1.6): Integer, RO, M
  - Hardware Address (e.g. MAC-Address)
- ifAdminStatus (2.1.7): Integer, RW, M
  - 1 .. up, 2 .. down, 3 .. testing (desired state)
- ifOperStatus (2.1.8): Integer, RO, M
  - 1 .. up, 2 .. down, 3 .. testing (actual state)
- ifLastChange (2.1.9): Integer, RO, M
  - System-time when interface become active
- ifInOctets (2.1.10): Counter, RO, M
  - Total number of received octets

Variables of the "Interface" Group

- ifInDiscards (2.1.13): Counter, RO, M
  - Total number of rejected input packets (because of input buffer overflow)
- ifInErrors (2.1.14): Counter, RO, M
  - Total number of received damaged packets
- ifOutOctets (2.1.16): Counter, RO, M
  - Total number of sent octets
- ifOutDiscards (2.1.19): Counter, RO, M
  - Total number of rejected output packets (because of output buffer overflow)
- ifOutErrors (2.1.20): Counter, RO, M
  - Total number of not-sent packets (because of errors)

Meaning of the Remaining Groups

- **Group "Address Translation"**
  - Mapping between network layer address (IP-address) and physical address
- **Croup "IP"**
  - IpAddrTable ... assigns IP-addresses, subnet-masks, and broadcast-methods to physical network interfaces (RO)
  - IpRouteTable ... contains Routing-table of this device (RW)
  - Counter for all IP-packets:
    - ipInReceives, ipInDiscard, ipInHdrError, ipInAddrError,
    - ipForwDatagrams, ipInDelivers, ipOutRequests,
    - ipOutDiscards,ipOutNoRoutes, ipFragOk, ipFragFails, etc...

Meaning of the Remaining Groups

- **Group "ICMP"**
  - Contains a counter for several packet-types and events
- **Group "TCP"**
  - Contains a counter for segments, parameters (e.g. retransmission timer value), session information (local and remote socket) and associated states, etc ...
- **Group "UDP"**
  - Contains a counter for UDP datagram's and information about currently involved connection endpoints (socket information)
- **Group "EGB"**
  - Contains a counter for EGP-messages and a table of neighbor routers

Agenda

- **Network Management Basics**
- **SNMP**
  - Basics
  - SMI
  - MIB
- **RMON**
- **SNMPv2**
- **Product Examples**

RMON

- **Monitoring of specific MIB variables**
  - Requires continuous polling of these variables
  - Doesn't replace functionality of a protocol analysator
- **Solution: Remote Monitoring (RMON)**
  - Defines network monitoring functions and communications between NMS and remote monitors
  - RMON agent supervises local segment autonomously
    - Delivers only results of statistics to the NMS
    - E.g. Traffic Matrix
  - RMON-MIB is a sub-tree below MIB-II
    - Using identifier 16

Agenda

- **Network Management Basics**
- **SNMP**
  - Basics
  - SMI
  - MIB
- **RMON**
- **SNMPv2**
- **Product Examples**

SNMPv1 Restrictions

- **Not suitable for large networks**
  - Polling efforts
  - Consider WAN links
- **Not efficient for carrying large data volumes**
  - Such as routing tables
- **Traps are not acknowledged**
  - Agent cannot be sure if its alarm reached NMS
- **Authentication is transmitted in plain-text Community strings**
- **No manager-to-manager communication**
- **Solution: SNMPv2**

SNMPv2 – New PDUs

- **GetBulkRequest**
  - For transmission of large data volumes such as routing tables
- **InformRequest**
  - For MIB-communication between network managers (client applications)
  - Hereby supporting multiple management stations
- **Report**
  - Not yet defined (?)
- **Trap PDU structure conforms to the format of the other PDU's**

SNMPv2 – Features

- **SNMPv1 was only designed to be transported via UDP and IP**
- **SNMPv2 defines several "Transport Mappings" in order to utilize other transport protocols**
  - OSI Connectionless Transport Service (CLTS, RFC 1418)
  - IPX (RFC 1420)
  - AppleTalk DDP (RFC 1419)

Agenda

- **Network Management Basics**
- **SNMP**
  - Basics
  - SMI
  - MIB
- **RMON**
- **SNMPv2**
- **Product Examples**

Product Examples

- **HP OpenView**
  - Comprehensive management system for heterogenous IT-applications
  - http://www.ovforum.org
- **Sun – Solstice Domain Manager**
- **Asante Technologies – IntraSpection**
  - Clients use Java-enabled Web browsers
- **Cabletron Systems – Spectrum**
  - Automated fault resolution with Case-based reasoning (AI)
- **Novell – ManageWise**
- **Tivoli Systems - TME 10 NetView**

