SNMP

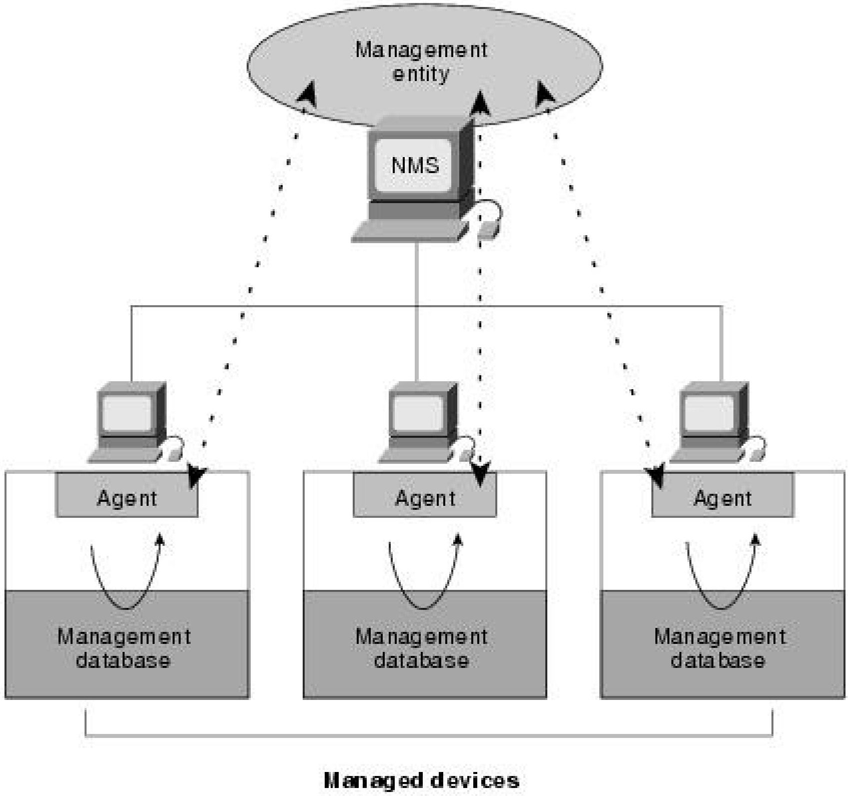
Simple Network Management Protocol

universidade de aveiro

deti.ua.pt

SNMP Basic Components

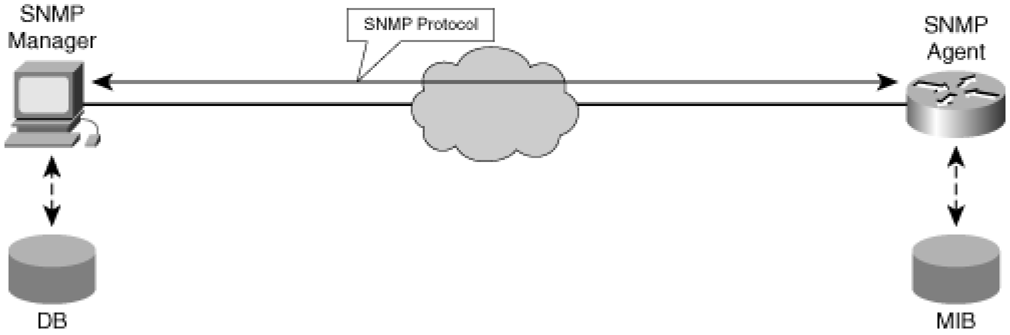
- An SNMP-managed network consists of three key components:
- Managed devices
  - Network node that contains an SNMP agent.
  - Collect and store management information and make this information available using SNMP.
  - Can be routers and access servers, switches and bridges, hubs, computer hosts, or printers.
- Agents
  - Network-management software module that resides in a managed device.
- Network-management systems (NMSs)
  - Executes applications that monitor and control managed devices.
  - Provide the bulk of the processing and memory resources required for network management.
  - One or more NMSs must exist on any managed network.



Data Collection Protocols: SNMP, SMI, and MIB

SNMP is an Internet protocol developed by the IETF

It is designed to facilitate the exchange of management information between network elements



- SNMP agent
  - A software module that resides in network elements; it collects and stores management information specified in the supported MIB modules. The SNMP agent responds to SNMP requests from an NMS station for information and actions. The SNMP agent can send fault notifications pro-actively to the SNMP manager.
- Managed object
  - A representation of something that can be managed.
  - Managed objects differ from variables, which are particular object instances.
- Management Information Base (MIB)
  - A collection of managed objects residing in a virtual information store.
  - A collection of related managed objects is defined in a specific MIB module.
  - A MIB can be considered a local data store at the network element.
- Syntax notation
  - A language used to describe managed objects in a machine-independent format
  - SNMP-based management systems use a subset of the International Organization for Standardization's (ISO) Open System Interconnection (OSI) Abstract Syntax Notation 1 (ASN.1, International Telecommunication Union Recommendation X.208) to define both the packets exchanged by the management protocol and the objects that are to be managed.
- Structure of Management Information (SMI)
  - Defines the rules for describing management information (the MIB). The SMI is defined using ASN.1.

SNMP Basic Commands

- Managed devices are monitored and controlled using four basic SNMP commands: read, write, trap, and traversal operations.
  - The **read** command is used by an NMS to monitor managed devices. The NMS examines different variables that are maintained by managed devices.
  - The **write** command is used by an NMS to control managed devices. The NMS changes the values of variables stored within managed devices.
  - The **trap** command is used by managed devices to asynchronously report events to the NMS. When certain types of events occur, a managed device sends a trap to the NMS.
  - Traversal operations are used by the NMS to determine which variables a managed device supports and to sequentially gather information in variable tables, such as a routing table.

SNMP: Polling

- Manager periodically asks the agent for new information

☺ Advantage: Manager completely controls the equipment, and knows all network details

☹ Disadvantage: Delay between event and its entry in the system, and unnecessary communication overhead:

  - Slow polling, slow answer to the events
  - Quick polling, quick reaction, but large bandwidth wastage

SNMP: Traps

- There is an event → trap is sent
- Trap contains appropriate information
  - equipment name, time instant of event, type of event

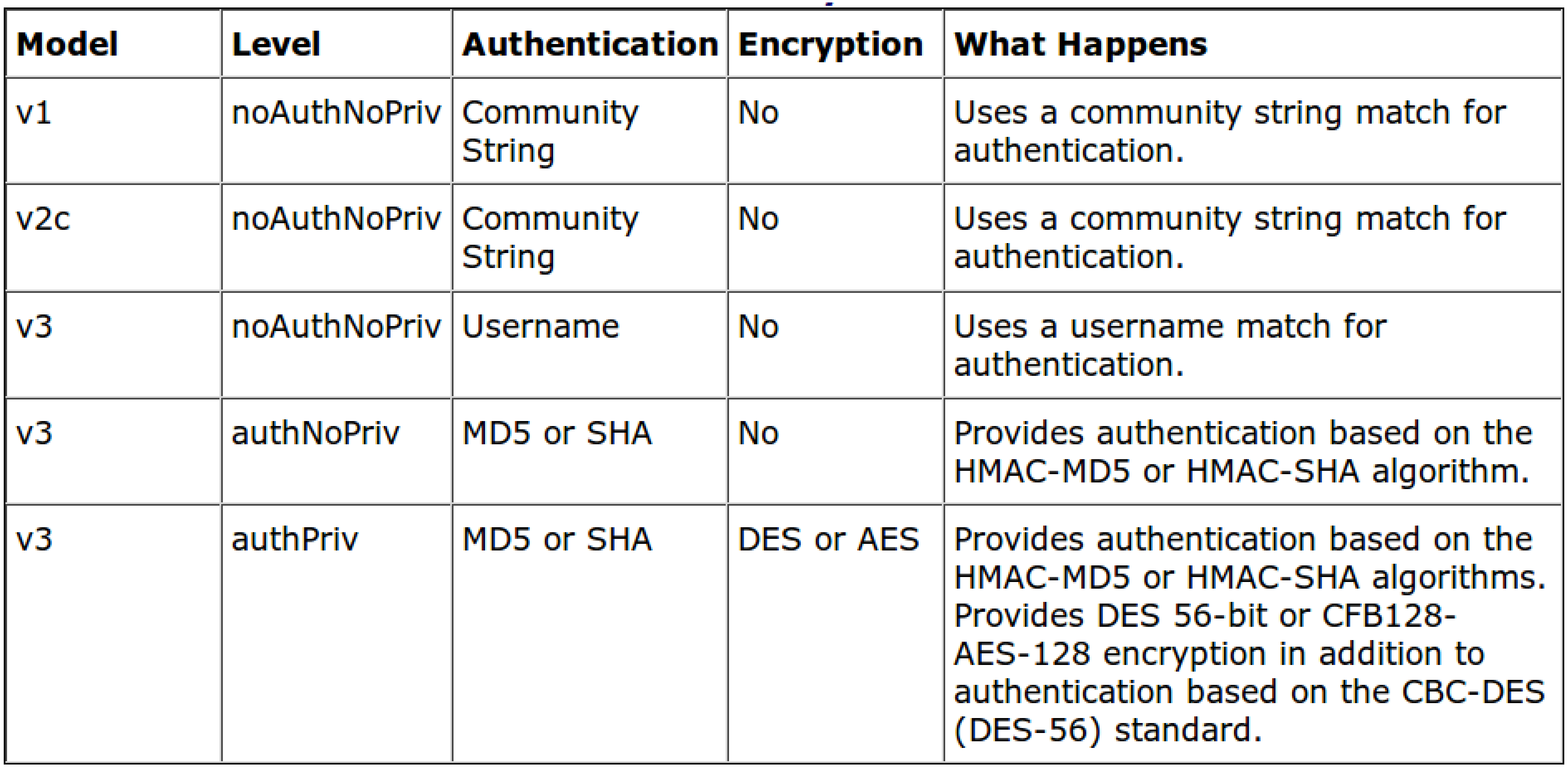
☺ Advantage: information only generated when required

☹ Disadvantage:

- ☹ More resources required in the managed equipment
- ☹ Traps can be useless
  - If many events occur, bandwitdh can be wasted with all traps (thresholds can solve)
  - Since the agent has only a limited scope of the network, NMS may already know about the events.

- Traps&Polling
  - Event occurs → trap is sent
  - Manager performs polling to obtain the rest of information
  - Manager also performs periodic polling, as backup

SNMP Versions



Model	Level	Authentication	Encryption	What Happens
v1	noAuthNoPriv	Community String	No	Uses a community string match for authentication.
v2c	noAuthNoPriv	Community String	No	Uses a community string match for authentication.
v3	noAuthNoPriv	Username	No	Uses a username match for authentication.
v3	authNoPriv	MD5 or SHA	No	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithm.
v3	authPriv	MD5 or SHA	DES or AES	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides DES 56-bit or CFB128-AES-128 encryption in addition to authentication based on the CBC-DES (DES-56) standard.

SNMPv1: security and authentication

- In its initial version, authorization and authentication were based on the notion of “SNMP community string”
- The “words of community” identify the permissions of the machine accessing the agent: read-only ou read-write
- By default, all systems are configured with the community strings:
  - public (read-only)
  - private (read-write)
- The words are case sensitive.

SNMPv2c and SNMPv3 versions

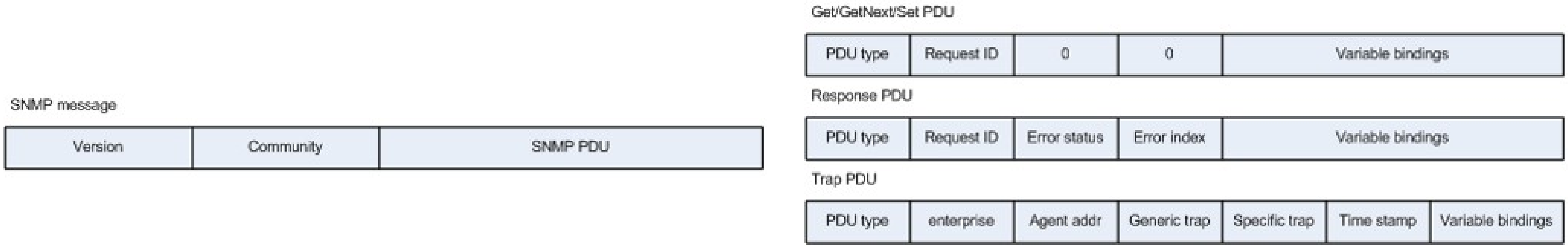
- SNMPv2 extensions
  - Structure of management information (SMI)
  - Manager-Manager capacity
  - New protocol operations
- SNMPv3 extensions
  - New message format
  - Message security
  - Access control

SNMPv3: Security

- Notion of “access control dependent on the user”
  - The agent mantains access rights information (policies) to different users in a data base
- Authentication: shared secret key
  - MD5 or SHA authentication passphrase hashes
- Privacy
  - Packet data may now be DES encrypted (future use allows additional encryption)
  - Passphrase defaults to authentication passphrase
  - Allows for unique Privacy passphrase
- Protection against replays: resort to nounces

SNMPv1 Message

- Version: SNMP version.
- Community: Community name, used for the authentication between an agent and the NMS.
  - In Get or GetNext operations, read community name is used for authentication;
  - In Set operation, write community name is used for authentication.
- Request ID: It is used to match a response to a request.
  - SNMP assigns a unique ID to each request.
- Error status: It is used in a response to indicate the errors when the agent processes the request
  - noError, tooBig, noSuchName, badValue, readOnly, and genErr.
- Error index: Provides the information of the variables that caused the error when an error occurs.
- Variable bindings: It is composed of a variable name and value.
- Enterprise: Type of the device that generates traps.
- Agent addr: Address of the device that generates traps.
- Generic trap: It includes coldStart, warmStart, linkDown, linkup, authenticationFailure, egpNeighborLoss and enterpriseSpecific.
- Specific trap: Specific trap information of a vendor.
- Time stamp: The amount of time between the time when the SNMP entity sending this message reinitialized and the time when traps were generated, that is, the value of sysUpTime.



SNMP message

Version	Community	SNMP PDU

Get/GetNext/Set PDU

PDU type	Request ID	0	0	Variable bindings

Response PDU

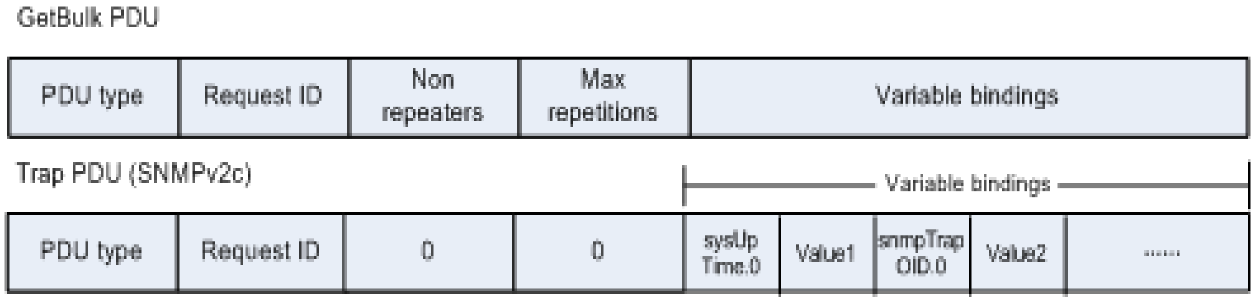
PDU type	Request ID	Error status	Error index	Variable bindings

Trap PDU

PDU type	enterprise	Agent addr	Generic trap	Specific trap	Time stamp	Variable bindings

SNMPv2c Message

- Compared with SNMPv1, GetBulk packets are added in SNMPv2c.
  - GetBulk operation corresponds to GetNext operation.
  - In a GetBulk operation, the setting of Non repeaters and Max repetitions parameters enables NMS to obtain data of many managed objects from an agent.
- In SNMPv2c, trap message format is different from that in SNMPv1.
  - SNMPv2c trap PDU adopts the format of SNMPv1 Get/GetNext/Set PDU, and sysUpTime and snmpTrapOID are used as variables in variable bindings to create a packet.



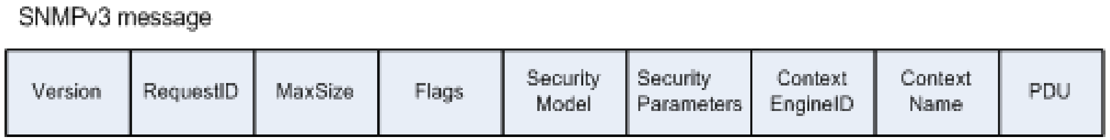
GetBulk PDU

PDU type	Request ID	Non repeaters	Max repetitions	Variable bindings

Trap PDU (SNMPv2c)

				Variable bindings				
PDU type	Request ID	0	0	sysUp Time.0	Value1	snmpTrap OID.0	Value2	......

SNMPv3 Message



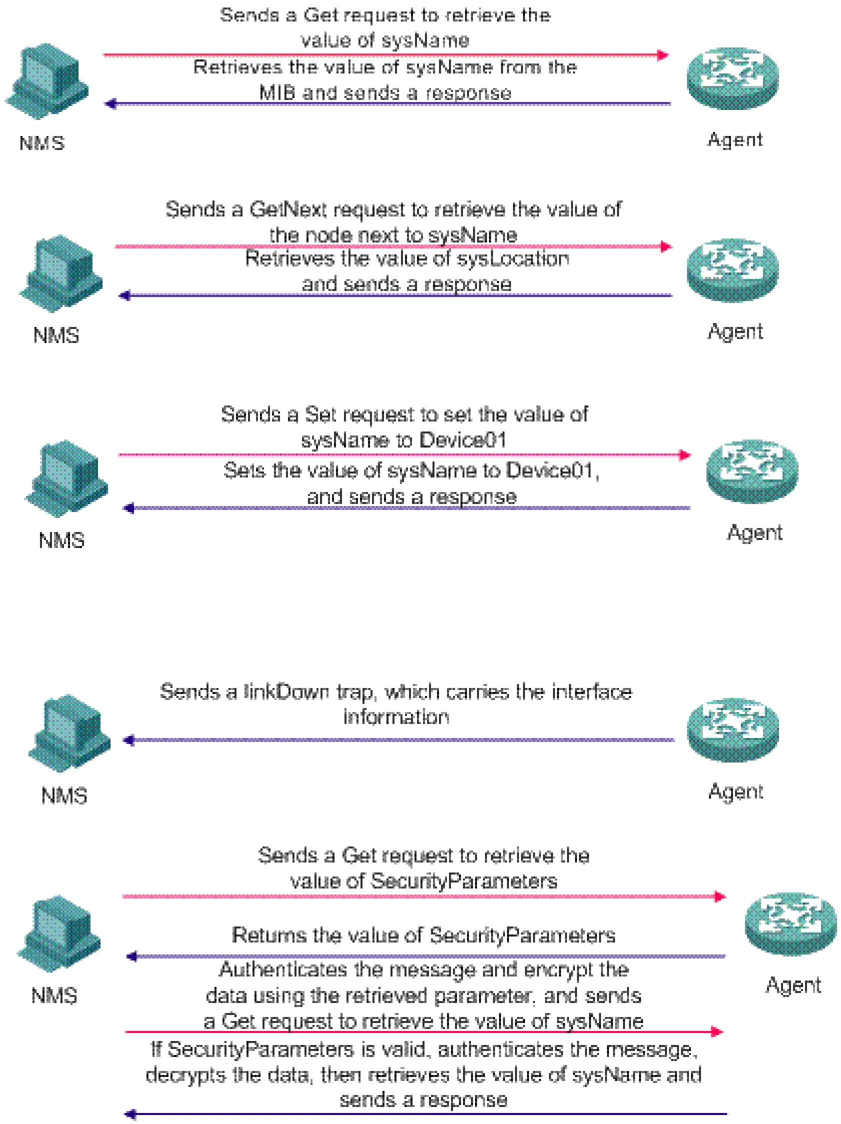
SNMPv3 message

Version	RequestID	MaxSize	Flags	Security Model	Security Parameters	Context EngineID	Context Name	PDU

- SNMPv3 message format is modified, but the PDU format is the same as that in SNMPv2c.
- The entire SNMPv3 message can be authenticated, and EngineID, ContextName, and PDU are encrypted.
- RequestID, MaxSize, Flags, SecurityModel and SecurityParameters form the SNMPv3 message header.
- Fields:
  - RequestID
  - MaxSize: The maximum size of the message that the sender of the message can receive.
  - Flags: Message flag which occupies one byte. Only the lowest three bytes are valid. 0x0 indicates no authentication no privacy, 0x1 indicates authentication without privacy, 0x3 indicates authentication with privacy, and 0x4 indicates to send a report PDU.
  - SecurityModel: Message security model, in the range 0 to 3. 0 indicates any model, 1 indicates SNMPv1 security model, 2 indicates SNMPv2c security model, and 3 indicates SNMPv3 security model.
  - SecurityParameters includes the following fields:
    - AuthoritativeEngineID: Specifies the snmpEngineID of the authoritative SNMP engine involved in the exchange of the message, used for identification, authentication and encryption for an SNMP entity. This field refers to the source for a trap, response, or report, and to the destination for a Get, GetNext, GetBulk, or Set operation.
    - AuthoritativeEngineBoots: Specifies the snmpEngineBoots value at the authoritative SNMP engine involved in the exchange of the message. It indicates the number of times that this SNMP engine has initialized or reinitialized itself since its initial configuration.
    - AuthoritativeEngineTime: Specifies the snmpEngineTime value at the authoritative SNMP engine involved in the exchange of the message. It is used for time window check.
    - UserName: Specifies the user (principal) on whose behalf the message is being exchanged. Usernames configured on NMS and Agent must be the same.
    - AuthenticationParameters: A key used in authentication calculation. If no authentication is performed, this field is null.
    - PrivacyParameters: A parameter used in privacy calculation.
  - ContextEngineID: Uniquely identifies an SNMP entity. For a message received, this field decides how this message will be processed; for a message sent, this field is provided by the sender.
  - ContextName: Identifies a context. Must be unique within an SNMP entity.

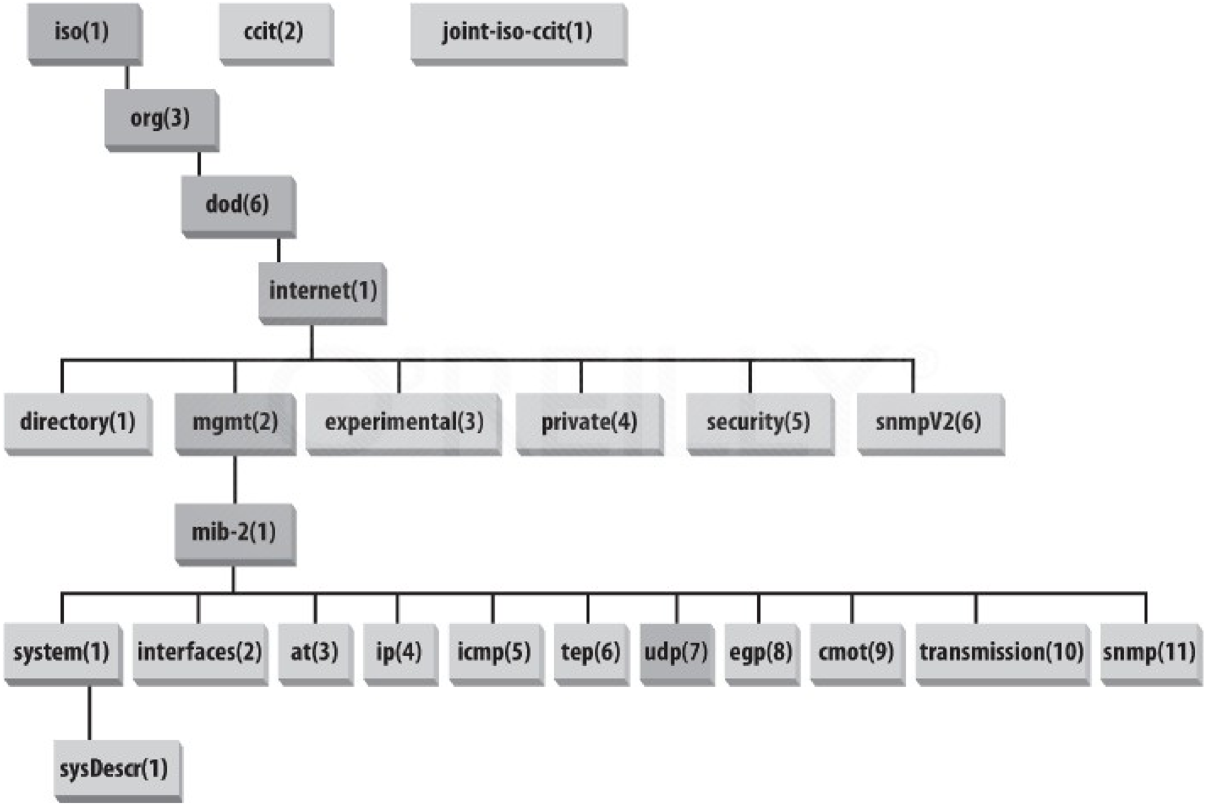
SNMP Operations

- SNMP provides the following five basic operations:
  - Get operation
    - Request sent by the NMS to the agent to retrieve one or more values from the agent.
  - GetNext operation
    - Request sent by the NMS to retrieve the value of the next OID in the tree.
  - Set operation
    - Request sent by the NMS to the agent to set one or more values of the agent.
  - Response operation
    - Response sent by the agent to the NMS.
  - Trap operation
    - Unsolicited response sent by the agent to notify the NMS of the events occurred.
- In SNMPv3 get operations are performed using authentication and encryption.


Sends a Get request to retrieve the value of sysName
Retrieves the value of sysName from the MIB and sends a response
NMS
Agent
Sends a GetNext request to retrieve the value of the node next to sysName
Retrieves the value of sysLocation and sends a response
NMS
Agent
Sends a Set request to set the value of sysName to Device01
Sets the value of sysName to Device01, and sends a response
NMS
Agent
Sends a linkDown trap, which carries the interface information
NMS
Agent
Sends a Get request to retrieve the value of SecurityParameters
Returns the value of SecurityParameters
Authenticates the message and encrypt the data using the retrieved parameter, and sends a Get request to retrieve the value of sysName
If SecurityParameters is valid, authenticates the message, decrypts the data, then retrieves the value of sysName and sends a response
NMS
Agent

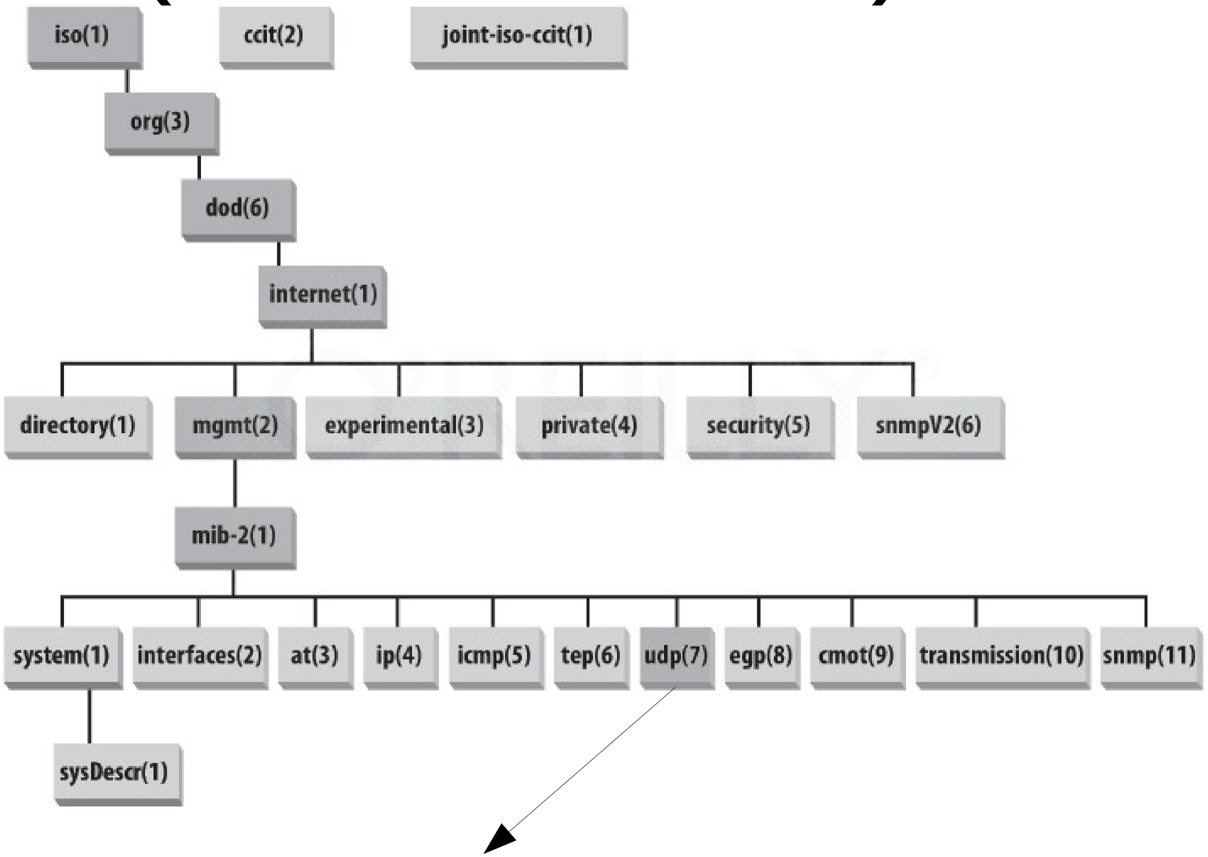
MIB Modules and Object Identifiers

- An SNMP MIB module is a specification of management information on a device
- The SMI represents the MIB database structure in a tree form with conceptual tables, where each managed resource is represented by an object
- Object Identifiers (OIDs) uniquely identify or name MIB variables in the tree
  - Ordered sequence of nonnegative integers written left to right, containing at least two elements
  - For easier human interaction, string-valued names also identify the OIDs
    - MIB-II (object ID 1.3.6.1.2.1)
    - Cisco private MIB (object ID 1.3.6.1.4.1.9)
- The MIB tree is extensible with new standard MIB modules or by experimental and private branches
  - Vendors can define their own private branches to include instances of their own products



SNMP Names (numbers/OID)

- To nominate all possible objects (protocols, data, etc.) it is used an ISO Object Identifier (OID) tree:
  - Hierarchic nomenclature of objects
  - Each leaf of the tree has a name and number



1.3.6.1.2.1.7.1

ISO

ISO-ident. Org.

US DoD

Internet

udpInDatagrams

UDP

MIB2

management

SNMP MIBs

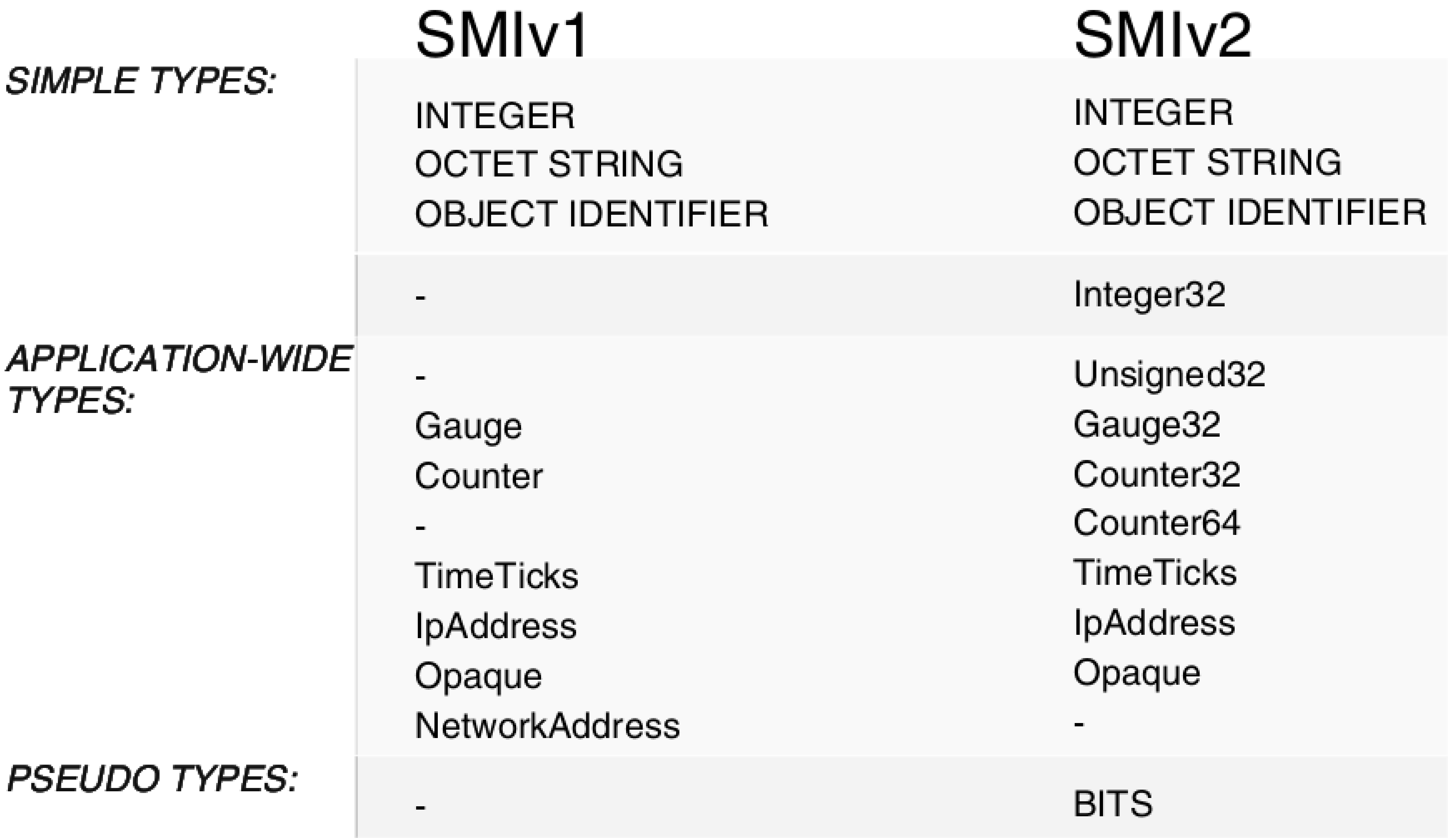
- Management Information Base (MIB): set of managed objects, used to define information from equipments, and created by the manufacturer
- Example: UDP module

Object ID	Name	Type	Comments
1.3.6.1.2.1.7.1	UDPInDatagrams	Counter32	Number of UDP datagrams delivered to users.
1.3.6.1.2.1.7.2	UDPNoPorts	Counter32	Number of received UDP datagrams for which there was no application at the destination port.
1.3.6.1.2.1.7.3	UDPInErrors	Counter32	The number of received UDP datagrams that could not be delivered for reasons other than the lack of an application at the destination port.
1.3.6.1.2.1.7.4	UDPOutDatagrams	Counter32	The total number of UDP datagrams sent from this entity.

SMI: Data language definition

- Well-defined sintax and semantics of management information
  - Type of basic data
    - INTEGER, Integer32, Unsigned32, OCTET, STRING, OBJECT IDENTIFIED, IPaddress, Counter32, Counter64, Guage32, Tie Ticks,Opaque...
  - Type of object
    - Type of data, status, semantic of the managed object
  - Module identification
    - Collection of objects inter-related in the MIB

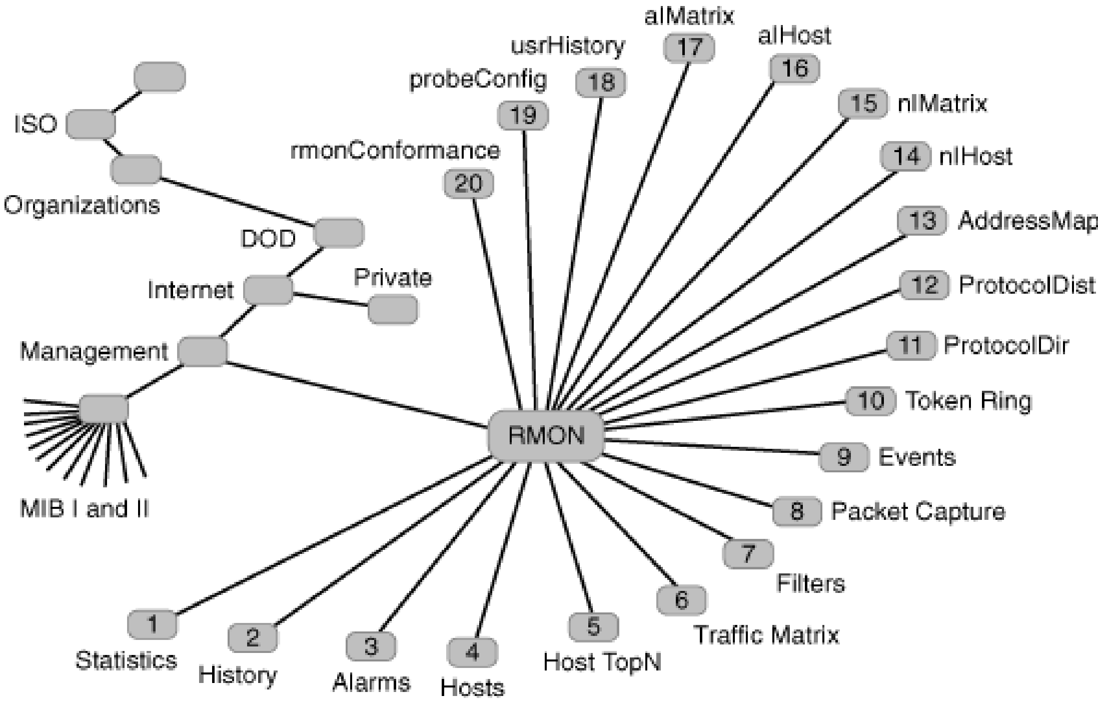
SMI: Data Types for Scalars



	SMIv1	SMIv2
SIMPLE TYPES:	INTEGER	INTEGER
	OCTET STRING	OCTET STRING
	OBJECT IDENTIFIER	OBJECT IDENTIFIER
	-	Integer32
APPLICATION-WIDE TYPES:	-	Unsigned32
	Gauge	Gauge32
	Counter	Counter32
	-	Counter64
	TimeTicks	TimeTicks
	IpAddress	IpAddress
	Opaque	Opaque
	NetworkAddress	-
PSEUDO TYPES:	-	BITS

RMON

- RMON is a set of standardized MIB variables that monitor networks
  - All previously defined MIBs monitored only nodes
- RMON has 9 groups
  - Statistics, History, Alarm, Host, HostTopN, Matrix, Filter, Packet Capture, and Event
- The term RMON now is often used to refer to the concept of remote monitoring and to the entire series of RMON MIB extensions
- The main RMON MIB extensions are:
  - RMON 1 and RMON 2 MIBs - Remote Monitoring MIB versions 1 and 2
  - DSMON MIB - Remote Monitoring MIB Extensions for Differentiated Services
  - SMON MIB - Remote Network Monitoring MIB Extensions for Switched Networks
  - APM MIB - Application Performance Measurement MIB


ISO
Organizations
DOD
Internet
Private
Management
MIB I and II
RMON
rmonConformance
20
probeConfig
19
usrHistory
18
alMatrix
17
alHost
16
15 nlMatrix
14 nlHost
13 AddressMap
12 ProtocolDist
11 ProtocolDir
10 Token Ring
9 Events
8 Packet Capture
7
Filters
6
Traffic Matrix
5
Host TopN
4
Hosts
3
Alarms
2
History
1
Statistics