Gestão & Segurança de Redes

MESTRADO EM ENGENHARIA INFORMÁTICA

Universidade do Minho
Departamento de Informática

NETWORK MANAGEMENT FOUNDATIONS

The need for standards...

- Heterogeneity of network devices & services.
- Too many communication protocols on network devices...
- Exponential growth of network devices, services and distributed applications.
- Deployment of configuration & quality control systems for network services.
- To not depend too much on human network managers...
- Deployment of accounting and contract service agreements.
- Deployment of external auditing systems.
- Deployment of management automation.

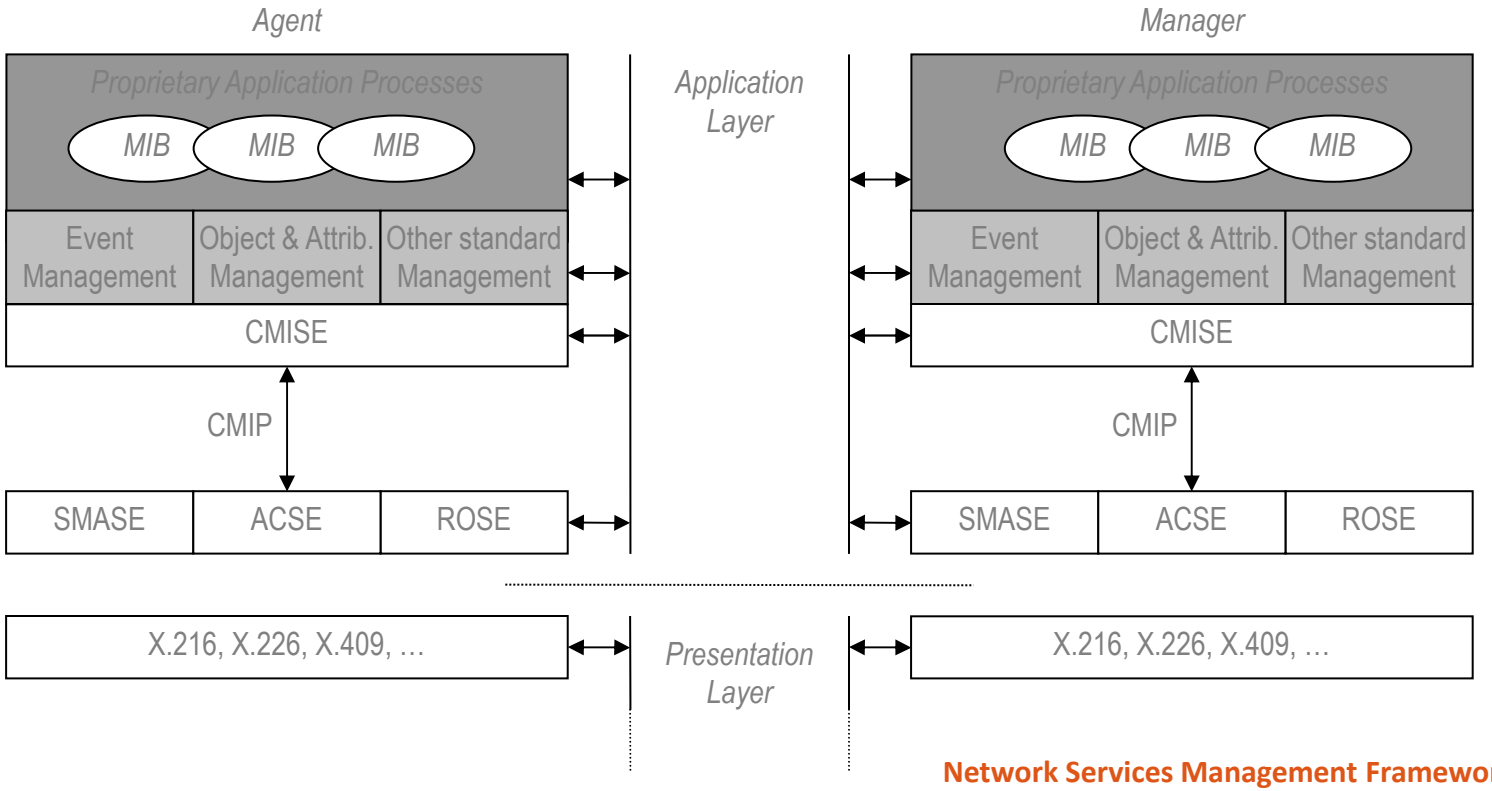
NETWORK MANAGEMENT FOUNDATIONS
TMN Architecture (ISO ITU-T M.3010)

- Exclusive management of telecommunications networks.
- Based on the ISO/OSI management functional model.
- It uses a dedicated data management communications network.
- Centralized architecture but with more distributed features than the ISO/OSI management architecture.

NETWORK MANAGEMENT FOUNDATIONS
OSI Management Architecture (X.700)

- Five functional areas (FCAPS): faults, configuration, accounting, performance and security.
- Management activity is also an application activity. All management entities need to implement the complete ISO/OSI protocol stack.
- Management Information Bases (MIBs) contain management objects that are abstractions of all managed resources.
- Heavily centralized system (poor scalability).
- Protocol/Interface Service: CMIP/CMIS.

NETWORK MANAGEMENT FOUNDATIONS
ISO/OSI Architecture



Network Services Management Framework
B.Dias, *PhD Thesis*
Universidade do Minho, December 2004.

NETWORK MANAGEMENT FOUNDATIONS
FCAPS Definition for Management Activities*

- **F**ault Management
- **C**onfiguration Management
- **A**ccounting Management
- **P**erformance Management
- **S**ecurity Management

Defined by the **International Engineering Consortium*

NETWORK MANAGEMENT FOUNDATIONS

FCAPS: Fault Management

- Diagnostic Testing
- Fault Detection/Isolation/Network Monitoring
- Fault Correction/Network Recovery
- Alarm Generation/Filtration/Handling/Correlation
- Logging & Statistics

NETWORK MANAGEMENT FOUNDATIONS

FCAPS: Configuration Management

- Resource Management (Initialization & Provisioning)
- Network & Services Discovering
- Configuration Policies Management & Automation
- User/Clients Management (Registration & Support)
- Logging & Statistics

NETWORK MANAGEMENT FOUNDATIONS

FCAPS: Accounting Management

- Resource Management
  (Costs Definition & Resource Usage)
- Users/Clients Quotas Monitoring, Reporting & Billing
- Auditing
- Logging & Statistics

NETWORK MANAGEMENT FOUNDATIONS
FCAPS: Performance Management

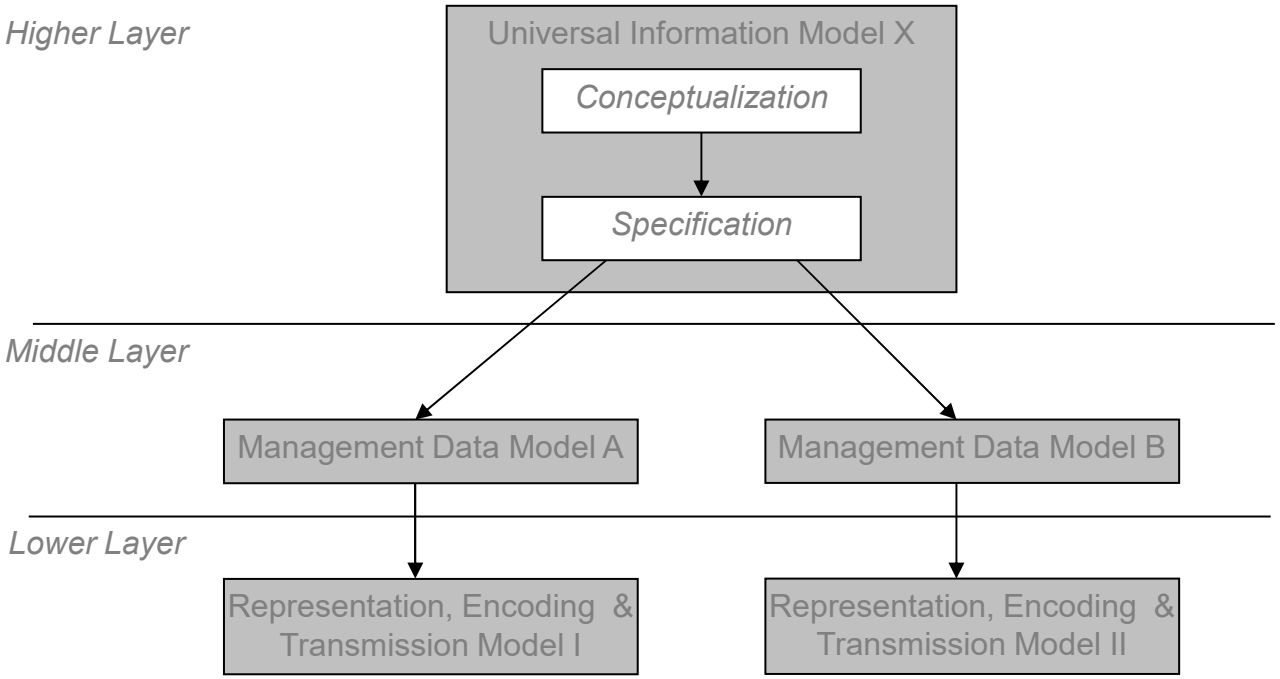
- Resource Utilization & Performance Monitoring (for network devices, systems and services)
- Users/Clients Utilization & Satisfaction
- Data Analysis & Capacity Planning
- Logging & Statistics

NETWORK MANAGEMENT FOUNDATIONS

FCAPS: Security Management

- Threat Management
  (Definition & Monitoring)
- Users/Clients Access Management & Certification
  (Definition, Monitoring & Reporting)
- Security Guarantees
  (Privacy, Authentication, etc)
- Auditing
- Logging, Data Analysis & Statistics

NETWORK MANAGEMENT FOUNDATIONS
Information & Data Management Models



Network Services Management Framework
B.Dias, *PhD Thesis*
Universidade do Minho, December 2004.

NETWORK MANAGEMENT FOUNDATIONS
Internet Network Management Framework (INMF)

- Simple management objects and communication protocol.
- Low consumption of resources on the managed devices.
- Simple and centralized architecture.
- Objects on Management Information Bases are based on the OSI MIB objects concept.
- Management services (either on agents or on managers) are application level services.
- Added security mechanisms on last versions.

NETWORK MANAGEMENT FOUNDATIONS

INMF: Historic Perspective

- Firstly, only the **Simple Network Management Protocol** (SNMP) was created, based directly on the **Simple Gateway Management Protocol** (SGMP).
- Other protocol alternatives at the time were refused:
  - > **CMIP over TCP** (CMOT);
  - > **High-Level Entity Management System** (HEMS).
- Three major versions of the framework:
  - > INMFv1, 1990-1992.
  - > INMFv2, 1993; Revised 1996.
  - > INMFv3,1999; Revised 2002-2003.

NETWORK MANAGEMENT FOUNDATIONS

INMF: Standard Components

> **Structure of Management Information** (SMI)
> **Management Information Bases** (MIBs)
> **Simple Network Management Protocol** (SNMP)
> **User-based Security Model** (USM)
> **View Access Control Model** (VACM)

- Communications Model is asynchronous and asymmetric.
- Monitoring system uses intensive polling of MIB variables.
- Identification of objects/variables and their instances is made through Object Identification (OID) values.

NETWORK MANAGEMENT FOUNDATIONS

INMF: Management Objects

- Types of management objects are defined on the SMI standard, which is a subset of the Abstract Syntax Notation 1 (ASN.1).
- Object types are simple and their manipulation/organization is functionally limited adding complexity to the managers implementation.
- Objects are conceptual abstractions of the managed devices/services/resources.
- Universal and hierarchical object identification is achieved with OIDs.
- Object grouping by function is made through MIB Groups.
- Access policies can be defined on MIB Views.

NETWORK MANAGEMENT FOUNDATIONS
INMF: Simple Network Management Protocol

- Application protocol for transport of the management information. Simple, asynchronous, asymmetric and *almost* non-confirmed.
- It is recommended to encapsulate SNMP on UDP, although other transport alternatives, like TCP, could be used (even encapsulation on lower layers of the TCP/IP stack).
- Four commands/primitives for managers: **snmp-get**, **snmp-getnext**, **snmp-getbulk** and **snmp-set**.
- Four commands/primitives for agents: **snmp-response**, **snmp-trap/notification** and **snmp-inform***.
- Few PDU format evolutions since SNMPv1.

NETWORK MANAGEMENT FOUNDATIONS
INMF: Security & Access Control

- Major evolution from SNMPv2 on.
- Complex mechanisms divided into two standards:
  > **User-based Security Model** (USM)
  (deployment of summation and encryption mechanisms)
  > **View Access Control Model** (VACM).
  (deployment of access control mechanisms)
- Recent and unbroken summation and encryption mechanisms should be used.
- There's no definition of a key concept or/and a distribution key mechanism.
- Current deployments may still use unsecure community strings!

NETWORK MANAGEMENT
INMF: Structure of Management Information

- Defines all possible types/syntaxes of management objects: SMIv1 (RFC1155) & SMIv2 (RFC2578).
- Each object definition is made up of three parts:
  > **Object Identifier** (OID)
  > ASN.1 Type/Syntax
  > (Implicit) network transmission coding using the **Basic Encoding Rules** (BER).
- Additional type definitions using **Textual Conventions**.
- Support declarations using **Conformance Statements**.

NETWORK MANAGEMENT

INMF: Structure of Management Information

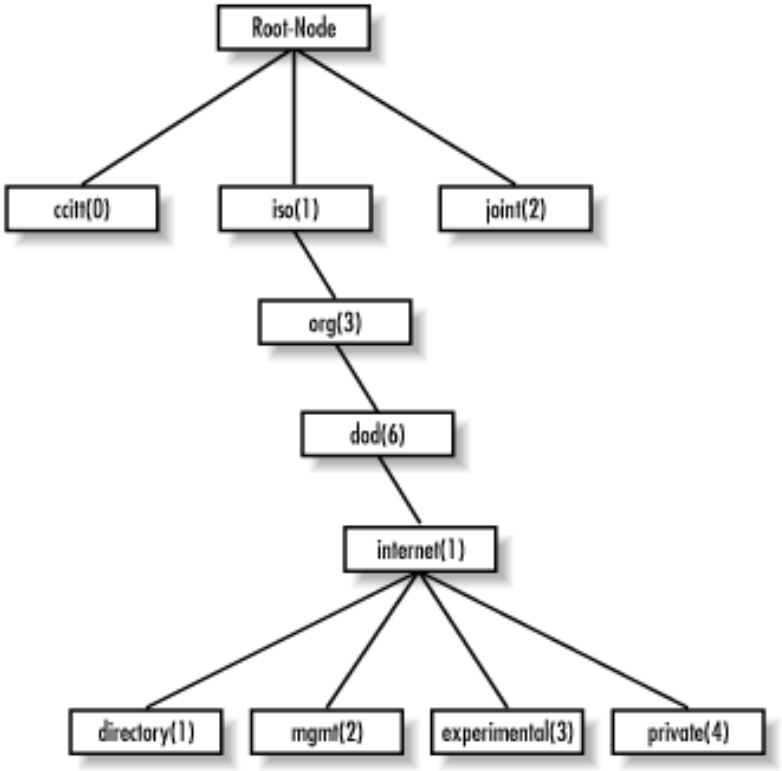
- Several **scalar object** types:
  > Octet String,
  > Bits, Unsigned, Integer,
  > Counter & Gauge (32 e 64 bits),
  > Timeticks,
  > Object Identifier,
  > NetAddress & IPAddress,
  > Opaque,
  > ...
- **Non-scalar objects** (for lists, tables, etc.):
  > Sequence of.

NETWORK MANAGEMENT
INMF: Structure of Management Information

- Some **Textual Conventions**:
  > DisplayString,
  > PhysAddress & MacAddress,
  > TruthValue & FalseValue,
  > TestAndInc,
  > TimeStamp, TimeInterval & DateAndTime,
  > StorageType,
  > VariablePointer,
  > TDomain & TAddress,
  > AutonomousType,
  > ...

NETWORK MANAGEMENT
INMF: Structure of Management Information

Hierarchical Object Identification



SNMP Essentials
D. Mauro, K. Schmidt
O'Reilly, 2001

NETWORK MANAGEMENT
INMF: Structure of Management Information

Hierarchical Object Identification:

```
[…]
internet       OBJECT IDENTIFIER ::= { iso(1) 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 }

[…]
```

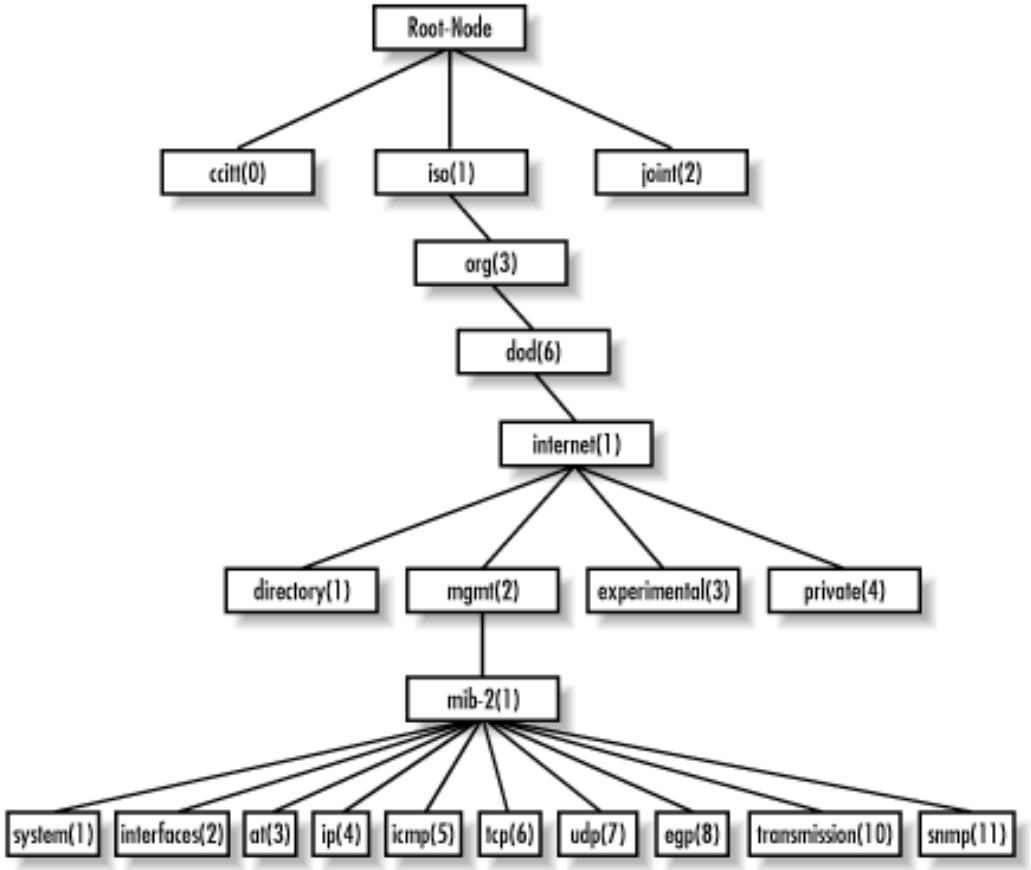
NETWORK MANAGEMENT
INMF: Management Information Bases

- One MIB standard (RFC 1213):
  > MIB-I (1990) ⇨ MIB-II (1991).
- One special MIB for statistical traffic monitorization on local area networks:
  > Remote Monitoring MIB (v2, RFC 2819).
- Many other MIBs, standards or not:
  > RFC 2863 -- Interfaces Group MIB
  > RFC 1850 -- OSPF Version 2 MIB
  > RFC 2790 -- Host Resources MIB
  > ...

NETWORK MANAGEMENT
INMF: Management Information Base II

```
RFC1213-MIB DEFINITIONS ::= BEGIN
          IMPORTS
                  mgmt, NetworkAddress, IpAddress, Counter, Gauge, TimeTicks FROM RFC1155-SMI
                  OBJECT-TYPE FROM RFC 1212;

          mib-2      OBJECT IDENTIFIER ::= { mgmt 1 }

-- groups in MIB-II

        system                        OBJECT IDENTIFIER ::= { mib-2  1 }
          interfaces                  OBJECT IDENTIFIER ::= { mib-2  2 }
          at                          OBJECT IDENTIFIER ::= { mib-2  3 }
          ip                          OBJECT IDENTIFIER ::= { mib-2  4 }
          icmp                        OBJECT IDENTIFIER ::= { mib-2  5 }
          tcp                         OBJECT IDENTIFIER ::= { mib-2  6 }
          udp                         OBJECT IDENTIFIER ::= { mib-2  7 }
          egp                         OBJECT IDENTIFIER ::= { mib-2  8 }
          transmission   OBJECT IDENTIFIER ::= { mib-2 10 }
          snmp                        OBJECT IDENTIFIER ::= { mib-2 11 }

-- the Interfaces table

          ifTable OBJECT-TYPE
              SYNTAX  SEQUENCE OF IfEntry
              ACCESS  not-accessible
              STATUS  mandatory
              DESCRIPTION
                  "A list of interface entries. The number of entries is
                   given by the value of ifNumber.“
              ::= { interfaces 2 }
[…]
```

NETWORK MANAGEMENT
INMF: Management Information Base II



SNMP Essentials
D. Mauro, K. Schmidt
O'Reilly, 2001

NETWORK MANAGEMENT
INMF: Simple Network Management Protocol

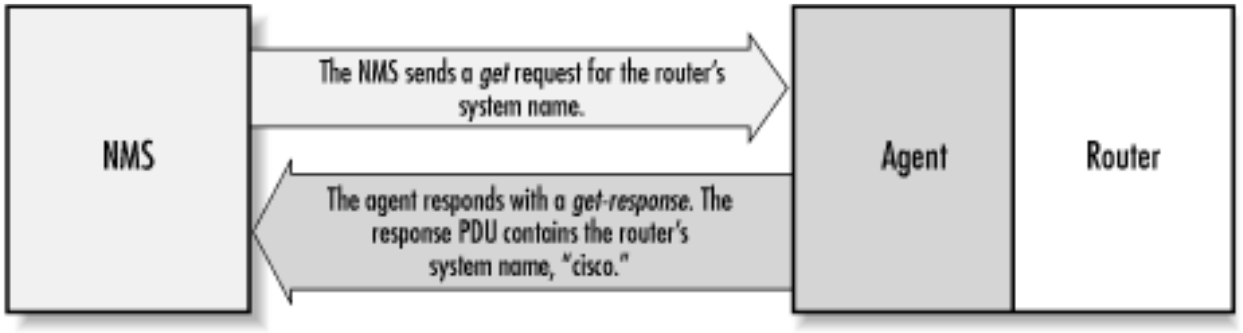
- Just two communications protocol versions:
  > SNMPv1 (RFC 1157) – INMFv1;
  > SNMPv2 (RFC 1905) – INMFv2 & INMFv3.

- Operations/Primitives (for *managers, **agents or ***both):
  > **get-req*** (SNMPv1 & v2)
  > **get-next-req*** (SNMPv1 & v2)
  > **get-bulk-req*** (SNMPv2)
  > **set-req*** (SNMPv1 & v2)
  > **inform-response*** (SNMPv2)
  > **get-response**** (SNMPv1 & v2)
  > **trap**** (SNMPv1) ⇨ **notification**** (SNMPv2)
  > **inform-req***** (SNMPv2)
  > **report***** (SNMPv2)

NETWORK MANAGEMENT
INMF: Simple Network Management Protocol

get-request()

```
$ snmpget –v2c –c public router-lab .1.3.6.1.2.1.1.5.0
system.sysName.0 = “cisco"
```



SNMP Essentials
D. Mauro, K. Schmidt
O'Reilly, 2001

NETWORK MANAGEMENT
INMF: Simple Network Management Protocol

getnext-request()

```
$ snmpwalk –v2c –c public router-lab system
system.sysDescr.0 = "Cisco Internetwork Operating [...]“
system.sysObjectID.0 = OID: enterprises.9.1.19
system.sysUpTime.0 = Timeticks:(27210723)3 days, 3:35:07.23
system.sysContact.0 = "“
system.sysName.0 = "cisco“
system.sysLocation.0 = “labcom-di-uminho-pt“
system.sysServices.0 = 6
```

Note: the Net-SNMP command `snmpwalk` is implemented using several getnext-request() primitives...

NETWORK MANAGEMENT
INMF: Simple Network Management Protocol

getbulk-request()

```
    $ snmpbulkget –v2c –c public -Cn1 Cr3 router-lab
  sysUpTime ifInOctets ifOutOctets
    system.sysUpTime.0 = Timeticks:(27210723) 3 days,3:35:07.23
    interfaces.ifTable.ifEntry.ifInOctets.1 = 70840
    interfaces.ifTable.ifEntry.ifOutOctets.1 = 70840
    interfaces.ifTable.ifEntry.ifInOctets.2 = 143548020
    interfaces.ifTable.ifEntry.ifOutOctets.2 = 111725152
    interfaces.ifTable.ifEntry.ifInOctets.3 = 0
    interfaces.ifTable.ifEntry.ifOutOctets.3 = 0
```

Note: –Cn option indicates *non-repeaters* parameter and –Cr indicates *max-repetitions* parameter of the getbulk-request() primitive...

NETWORK MANAGEMENT
INMF: Simple Network Management Protocol

set-request()

```
$ snmpget –v2c –c public router-ext sysLocation.0
system.sysLocation.0 = “labcom-di-uminho-pt”

$ snmpset –v2c –c labcompasswd router-ext labcom
sysLocation.0 s “Buraco Negro”
system.sysLocation.0 = “Buraco Negro”

$ snmpgetnext –v2c –c public router-ext sysLocation
system.sysLocation.0 = “Buraco Negro”
```

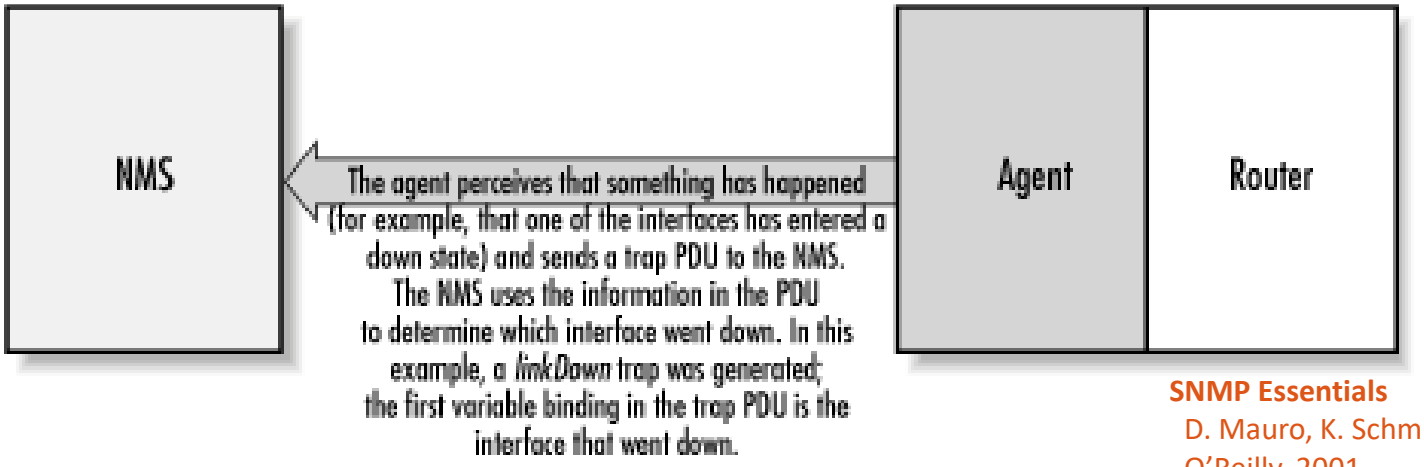
Note: the ‘s’ parameter indicates the type of the object...

NETWORK MANAGEMENT
INMF: Simple Network Management Protocol

trap()/notification()

Non-solicited information that agents send to managers, referring events that may need special treatment. No response from managers. Examples:

- Change in network interface status;
- Memory malfunction;
- Secure temperature threshold surpassed…



SNMP Essentials
D. Mauro, K. Schmidt
O'Reilly, 2001

NETWORK MANAGEMENT
INMF: Simple Network Management Protocol

trap()/notification()

ID of the “first” traps/notifications defined in SNMP:

```
<0> coldStart
<1> warmStart
<2> linkDown
<3> linkUp
<4> authorizationFailure
<5> egpNeighborLoss
<6> enterpriseSpecific
<…> …
```

NETWORK MANAGEMENT
INMF: Simple Network Management Protocol

inform-request()
Non-solicited information that agents or managers send to managers, referring events that may need special treatment and response from managers is expected. Until the manager confirms the reception of the inform request the sender should keep on sending the same inform request primitive (at least while the condition that generated the inform request is maintained).

report()
Non-solicited information that agents or managers send to agents or managers, referring events about the SNMP functionality that may need special attention from other management elements. There's no response expected. Introduced as experimental on SNMPv2 and standard on SNMPv3 is not generally used.

NETWORK MANAGEMENT
INMF: Simple Network Management Protocol

SNMP Error Codes

Examples of some SNMP Error Status Codes that must be included in the response primitive:

```
<0> noError*                <1> tooBig*
<2> noSuchName*             <3> badValue*
<4> readOnly*               <5> genErr*
<6> noAccess                <7> wrongType
<8> wrongLength             <9> wrongEncoding
<10> wrongValue             <11> noCreation
<12> inconsistentValue      <13> resourceUnavailable
<14> commitFalied           <15> undoFailed
<16> authorizationError     <17> notWritable
<18> inconsistentName
```

*SNMPv1

NETWORK MANAGEMENT
INMF: Simple Network Management Protocol

SNMPv1 & SNMPv2c Messages

All SNMP Messages, including v3, are defined in ASN.1 and coded and transmitted using the Basic Encoding Rules (BER).

SNMPv1*/SNMPv2c MESSAGE

Protocol Version	Version=0*/1	INTEGER

Community	Name	STRING
Non-Scoped PDU	**SNMP PDU**	

SNMP PDU

PDU Header	Type	INTEGER
	Request ID	INTEGER
	Error Status	INTEGER
	Error Index to VarBind List	INTEGER

VarBind List	Variable OID1	OID
	Variable Value1	…
	Variable OID2	OID
	Variable Value2	…
	…	…

NETWORK MANAGEMENT
INMF: Simple Network Management Protocol

SNMPv3 Messages

SNMPv3 MESSAGE

Protocol Version	Version=3	INTEGER
Message Header	Message ID	INTEGER
	Message Max. Size	INTEGER
	Flags	STRING
	Security Model=3(USM)	INTEGER
Security Parameters	Engine ID	INTEGER
	Engine Boots	INTEGER
	Engine Time	INTEGER
	User Name	STRING
	Authentication Parameters	STRING
	Privacy Parameters	STRING
Scoped PDU	Context Engine ID	STRING
	Context Name	STRING
	SNMP PDU	...

NETWORK MANAGEMENT
INMF: Defining Tables in MIBs

```
nameOfTheTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF TypeOfTheEntries
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION   "A description of the table."
    ::= { theGroup N }

nameOfTheVirtualEntry OBJECT-TYPE
    SYNTAX        TypeOfTheEntries
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION   "A description of each entry/row
                  of the table."
    INDEX { theKeyObjects } ::= { nameOfTheTable 1 }
```

NETWORK MANAGEMENT
INMF: Defining Tables in MIBs

```
TypeOfTheEntries ::=
    SEQUENCE {
        nameOfTheFirstObject   TypeOfTheFirstObject
        [...]
        nameOfTheLastObject    TypeOfTheLastObject
    }

nameOfTheFirstObject OBJECT-TYPE
    SYNTAX        TypeOfTheFirstObject
    MAX-ACCESS    read-only|read-write
    STATUS        current
    DESCRIPTION   "A description of the first
                  object/column."
    ::= { nameOfTheVirtualEntry 1 }
```

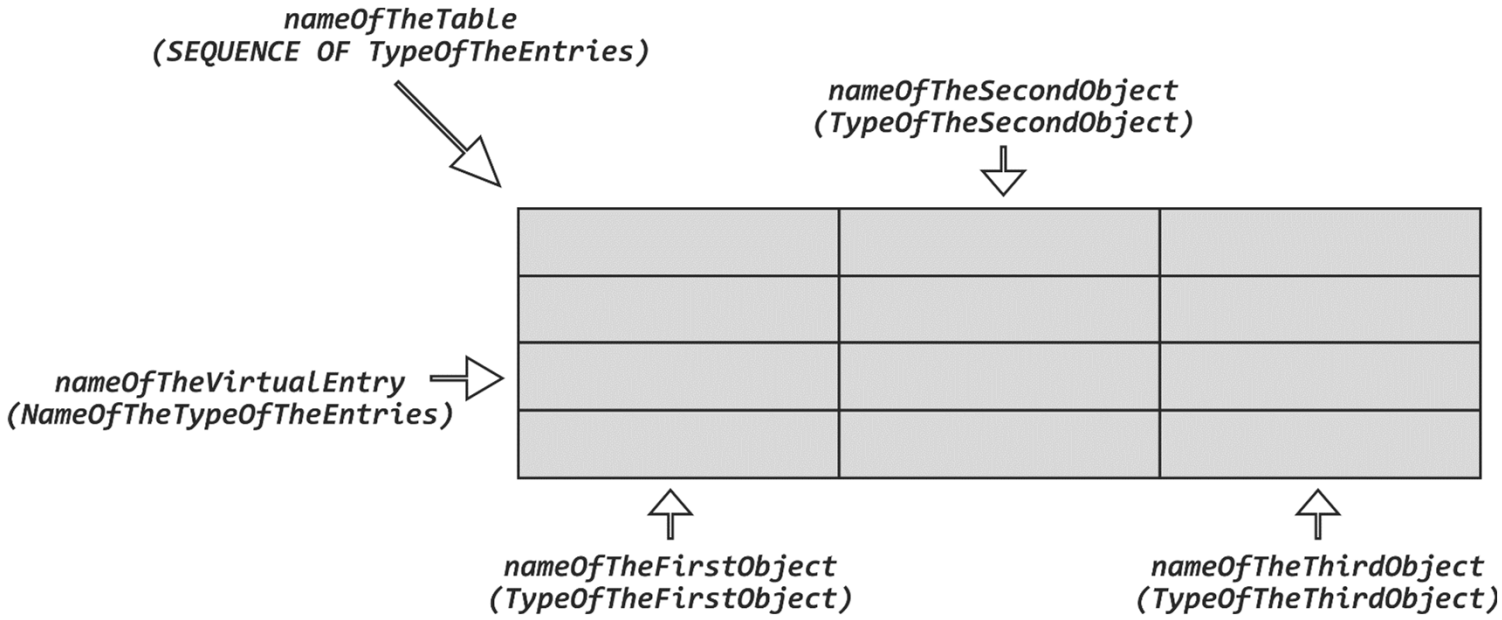
NETWORK MANAGEMENT
INMF: Defining Tables in MIBs

[...]

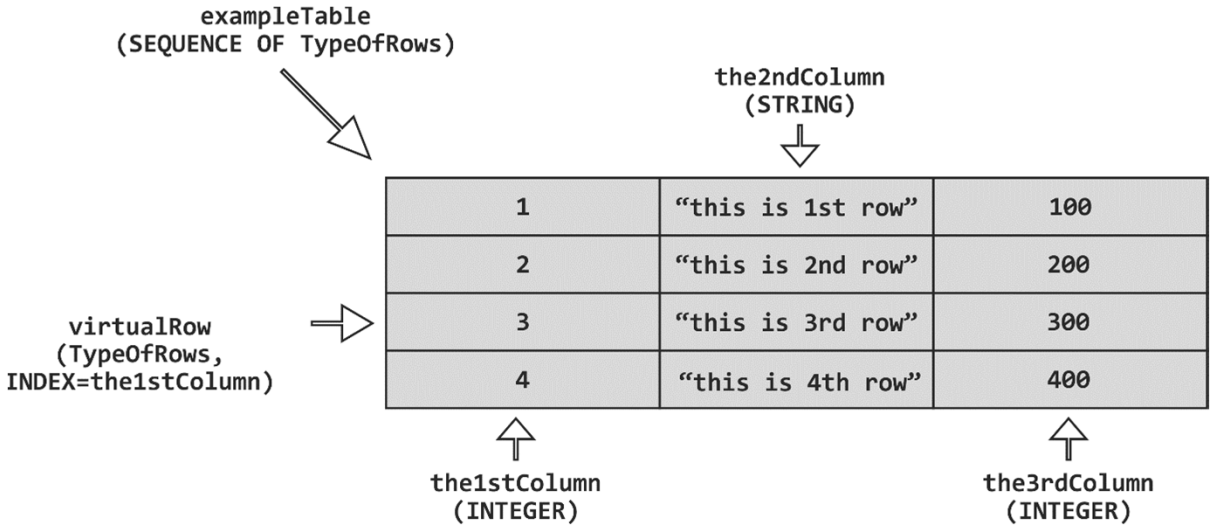
```
nameOfTheLastObject OBJECT-TYPE
    SYNTAX          TypeOfTheLastObject
    MAX-ACCESS      read-only|read-write
    STATUS          current
    DESCRIPTION     "A description of the last
                    object/column."
    ::= { nameOfTheVirtualEntry M* }
```

*For ***M*** objects/columns in each entry/row of the table.

NETWORK MANAGEMENT
INMF: Defining Tables in MIBs



NETWORK MANAGEMENT
INMF: Defining Tables in MIBs




```
> snmpget -v2c -c public 127.0.0.1 the2ndColumn.2 the3rdColumn.4
     the2ndColumn.2 = "this is 2nd row"
     the3rdColumn.4 = 400

> snmpgbulkget -v2c -c public -Cn0 -Cr2 127.0.0.1 the1stColumn the2ndColumn the3rdColumn
     the1stColumn.1 = 1
     the1stColumn.2 = 2
     the2ndColumn.1 = "this is 1st row"
     the2ndColumn.2 = "this is 2nd row"
     the3rdColumn.1 = 100
     the3rdColumn.2 = 200
```

NETWORK MANAGEMENT
INMF: Security & Access Control

Main Threads

- Impersonation or Masquerade: using the identity of others to perform unauthorized management operations.
- Modification of information: destruction/omission or modification of information in messages, including the type of the commands, or the entire SNMP messages.
- Disclosure of information: this includes any information contained in SNMP messages (commands, instance IDs and values, identities, errors, etc.) or information about the flow of messages (traffic analysis).
- Disruption of service: any type of behavior-oriented attack that may disrupt the agents or managers intended service levels, including Denial of Use or Denial of Service (DoS) attacks.

NETWORK MANAGEMENT
INMF: Security & Access Control

SNMPv1 & SNMPv2c

- No real security mechanisms (no encryption or authentication), which renders any real time access control features useless but simplifies implementation.
- Community Names and MIB Views help to define Access Policies but there's no secure deployment of them.
- A Community identifies a group of managers; a MIB View identifies a group of objects of one or more MIBs; an Access Mode of `read-only` or read-write can be associated to each MIB View, defining an Access Profile (or Community Profile); pairing an Access Profile with a Community defines an SNMP Access Policy.
- Agents should have means to configure Community Names, MIB Views, Access Profiles and Access Policies.

NETWORK MANAGEMENT
INMF: Security & Access Control

SNMPv2 & SNMPv3

- Real security mechanisms: authentication, data integrity verification and confidentiality, which complicates implementation, configuration and deployment.
- Authentication and data integrity verification is implicit using hash mechanisms with symmetric keys and confidentiality is explicitly attained by using symmetric key encryption mechanisms.
  Strategies and rules to implement these mechanisms are defined on the **User-based Security Model (USM)**.
- Control Access rules and mechanisms are defined on the **View-Access Control Model (VACM)**.

NETWORK MANAGEMENT
INMF: Security & Access Control

SNMPv2 & SNMPv3

- User Names (and their respective secrets) are used instead of Community Names.
- Agents and Managers should have means to configure and securely share symmetric keys (or secrets) but there's no standard for this.
- The USM recommends, as minimum requirements, the use of an HMAC method as the hash mechanism and DES as the encryption mechanism.
- The USM also defines three possible security modes for SNMPv3: `noAuthNoPriv`, `AuthNoPriv` and authPriv (`noAuthPriv` is, obsviously, not possible) although `noAuthNoPriv` should not be used as this mode has no security guarantees and is equivalent to the SNMPv1/v2c insecurity.