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.

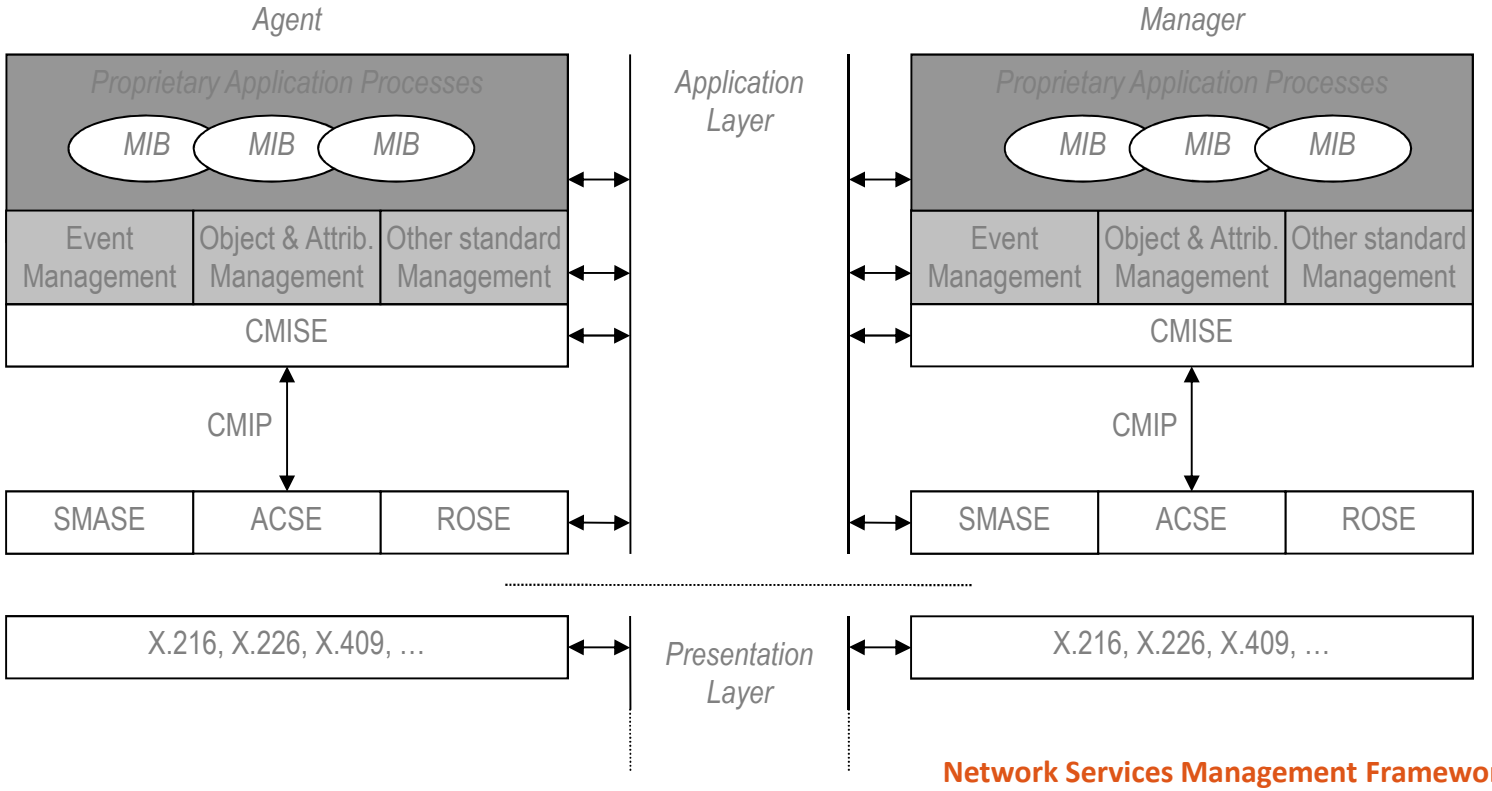
NETWORK MANAGEMENT FOUNDATIONS
TMN Architecture (ISO ITU-T M.3010)

- 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

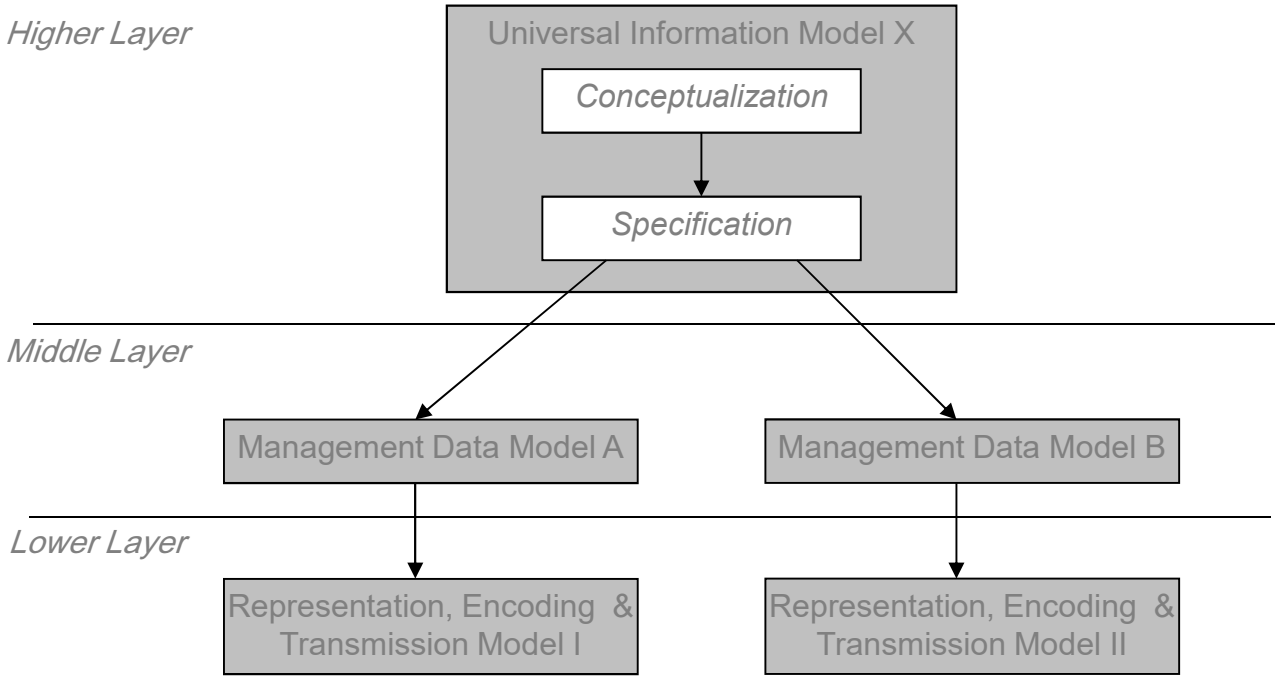
- Resource Utilization & Performance Monitoring (for network devices, systems and services)
- Users/Clients Utilization & Satisfaction
- Data Analysis & Capacity Planning
- Logging & Statistics

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

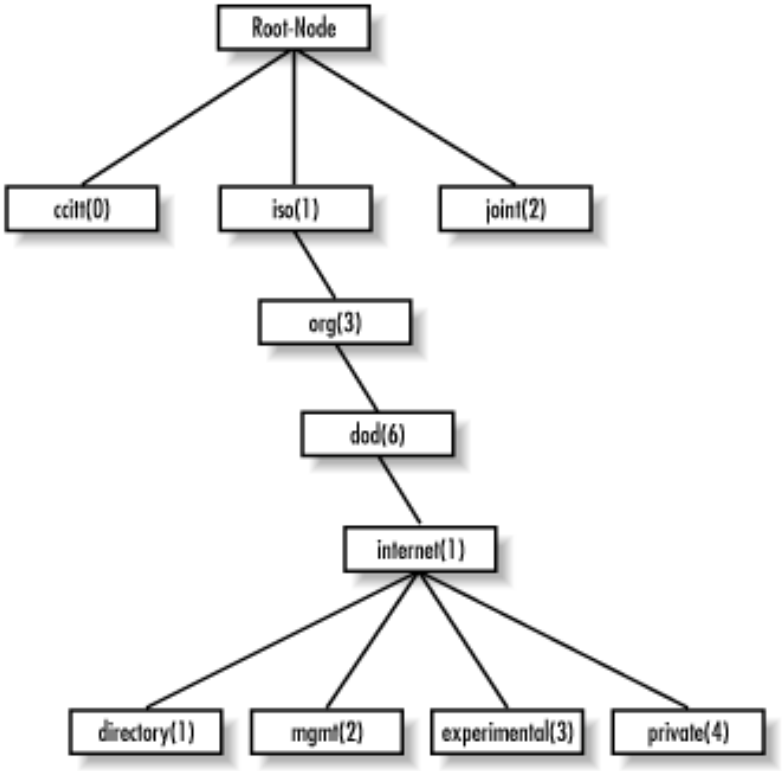
- Several **scalar object** types:
  > Octet String,
  > Bits, Unsigned, Integer,
  > Counter & Gauge (32 e 64 bits),
  > Timeticks,
  > Object Identifier,
  > NetAddress & IPAddress,
  > Opaque,
  > ...
- **Non-scalar objects** (for lists, tables, etc.):
  > Sequence of.

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 }

[…]
```

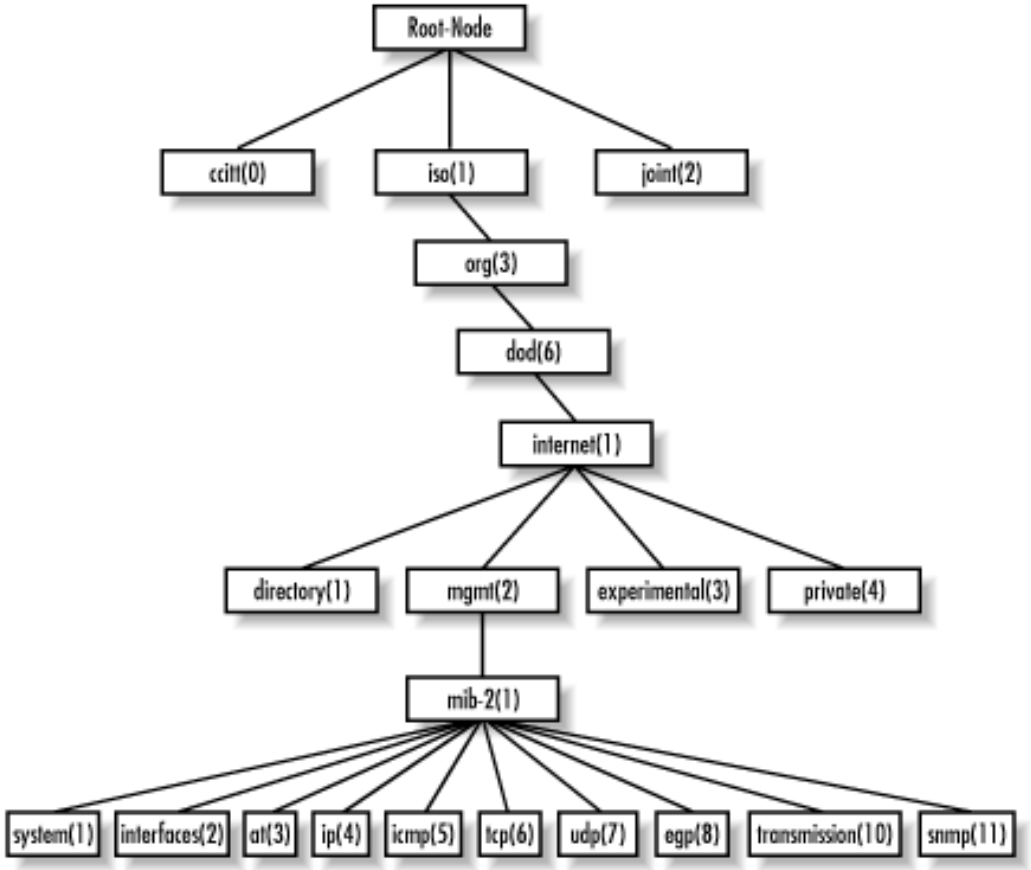
NETWORK MANAGEMENT
INMF: Management Information Bases

- 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

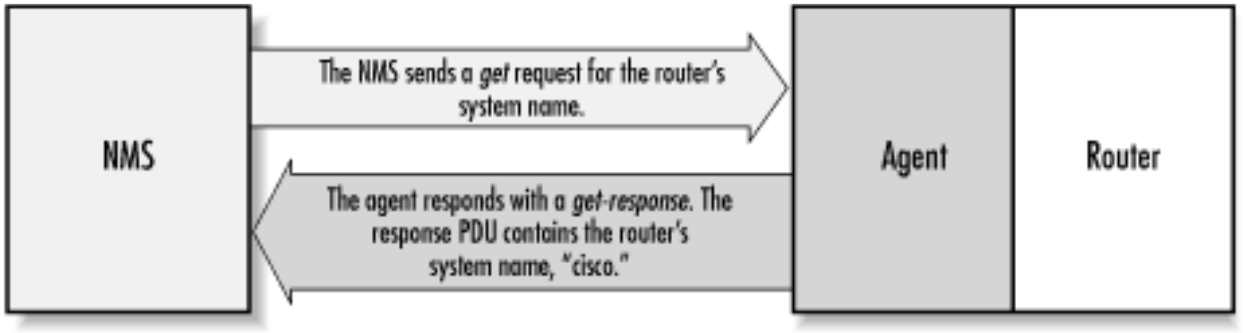
- Just two communications protocol versions:
  > SNMPv1 (RFC 1157) – INMFv1;
  > SNMPv2 (RFC 1905) – INMFv2 & INMFv3.

- 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”
```

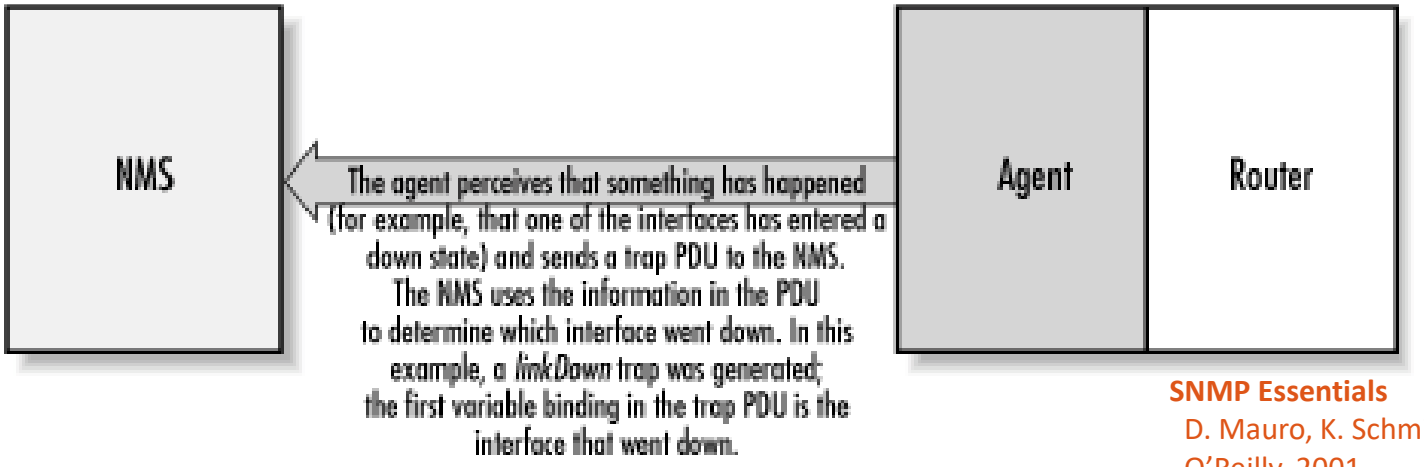
Note: the ‘s’ parameter indicates the type of the object...

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 }
```

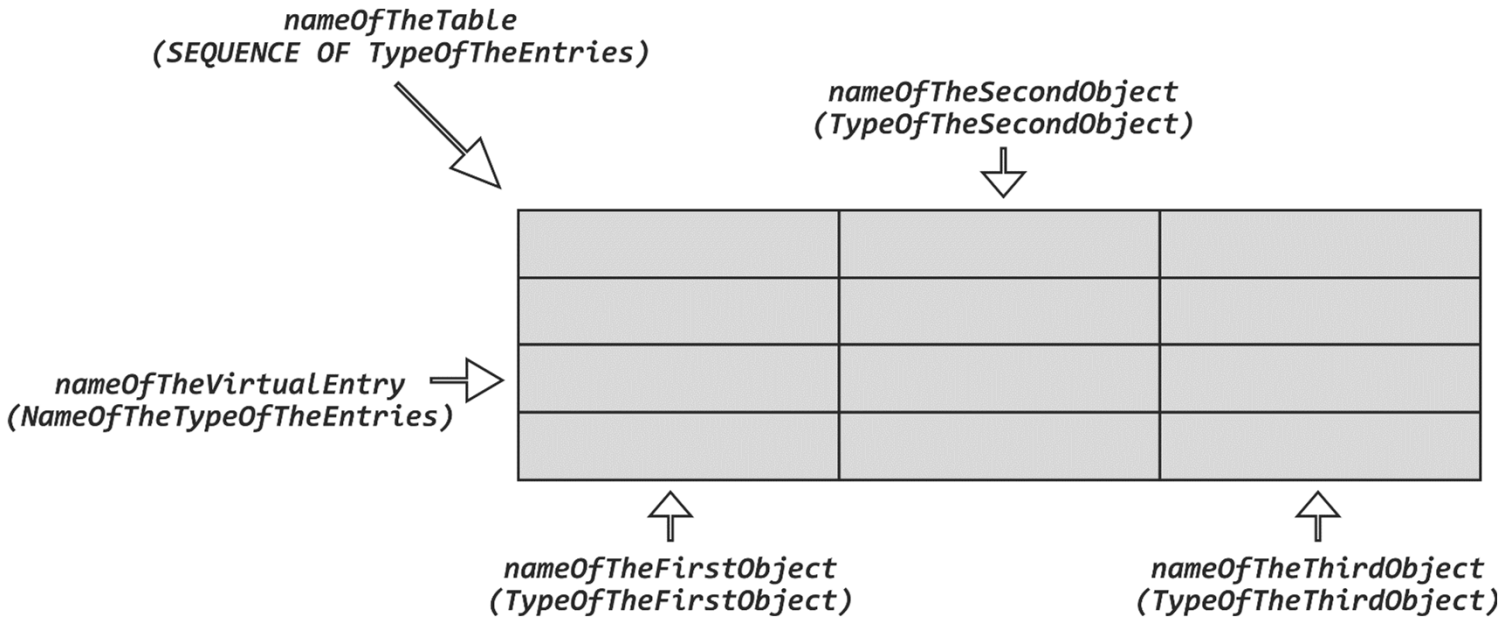
NETWORK MANAGEMENT
INMF: Defining Tables in MIBs

[...]

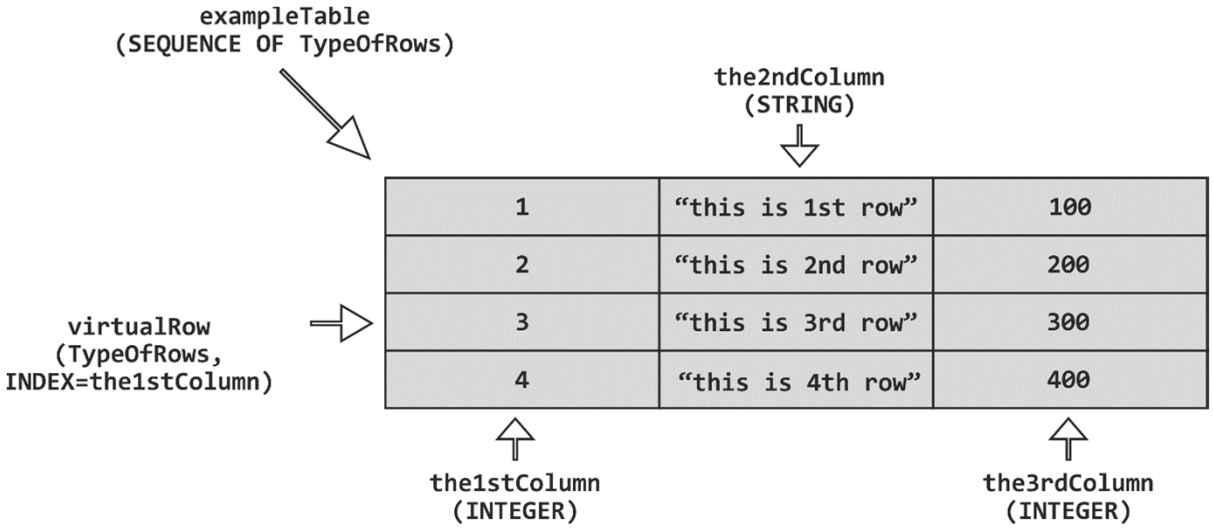
```
nameOfTheLastObject OBJECT-TYPE
    SYNTAX          TypeOfTheLastObject
    MAX-ACCESS      read-only|read-write
    STATUS          current
    DESCRIPTION     "A description of the last
                    object/column."
    ::= { nameOfTheVirtualEntry M* }
```

*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.

NETWORK MANAGEMENT
INMF: Advanced Mechanisms

Distributed Management

- Initial efforts were created on standard INMF working groups:
  - ✓ **RMON MIB** – for monitoring the Internet traffic (all layers) interfaces on a LAN (Remote Monitoring MIB).
  - ✓ **M2M MIB** – for added communication between managers (Manager to Manager MIB).
- Later, a special IETF working group was created, the DISMAN Group (IETF Distributed Management Working Group).

NETWORK MANAGEMENT
INMF: Advanced Mechanisms

DISMAN: Alarms

- This feature permits the remote configuration of management agents to implement alarms.
- It is a local process in the agent that is continuously monitoring the value of the instances of the MIB objects. This process compares these values and sets/triggers the associated alarms when set of pre-defined threshold values are attained.
- So, this process of monitoring and alarm setting/triggering is a management activity delegated by the manager to the agent.
- This concept was firstly introduced, in an implied form, on the RMON MIB.

NETWORK MANAGEMENT
INMF: Advanced Mechanisms

DISMAN: Events

- Events permit agents to inform, in an unsolicited way, managers about occurrences of pre-defined events.
- Events and alarms are complementing but also overlapping features as they are based, more or less, on the same concept. Both approaches could be united on a common feature. Although events convey a more generic concept there has been more developments on the alarm mechanism.
- Alarms and events generate notifications (or traps) that can be logged through a special MIB – the **Notifications Log MIB**.

NETWORK MANAGEMENT
INMF: Advanced Mechanisms

DISMAN: Delegation of Scripts

- This mechanism can be used to delegate management procedures from an SNMP manager to another SNMP entity (generally, an agent) that implements the referred code.
- The standard does not define or recommends any standard language or encoding technique for the script code.
- The standard also does not define a security model/system to be applied when dealing with the remote execution of management scripts.
- On the other hand, access control should be implemented using the VACM approach.

NETWORK MANAGEMENT
INMF: Advanced Mechanisms

DISMAN: Delegation of Scripts

- Mechanisms for identification of scripts (from a script repository) and their parameterization are implementation dependent and parameterization should be made offline.
- The most used methodologies either use a push method for transfer of the script code from the manager to the agent using normal MIB table manipulation techniques or a poll method from the agent which normally involves an HTTP or FTP access to an URL (passed by the manager to the agent) where the script code resides.
- It is not possible to re-use scripts.
- This type of distributed mechanism is complex to implement and the implementations are not universal.

NETWORK MANAGEMENT
INMF: Advanced Mechanisms

DISMAN: Other mechanisms...

- **Remote calculation of expressions** – this feature, defined on the **Expression MIB**, permits the computation of Boolean expressions for definition of alarm thresholds and event triggering conditions.
- **Scheduling of management procedures** – this feature can be used for scheduling of simple management procedures that can be abstracted by integer values; it is also possible to schedule management scripts if the management entity implementing the scheduler also implements the Script MIB. Scheduling of complex management procedures or conditional scheduling is not practical due to its implementation complexity so it is not conceptually supported.

NETWORK MANAGEMENT
INMF: Advanced Mechanisms

Policy Management

- **Configuration Management with SNMP, IETF Working Group.** It recommends methodologies/strategies to tackle the problem of configuration management using the regular SNMP architecture.
- **Network Management Research Group, IRTF Working Group.** This research group defined a variant of the INMF architecture that serves to create and manipulate management policies. This variant uses a higher-level modeling language, the **SMI for the Next Generation** (SMIng), to define policies and the **Structure of Policy Provisioning Information** (SPPI) is used to map SMIng specifications into **Policy Information Bases** (PIBs), which are the special MIBs used for definition of policy management objects. The SPPI standard is a sub-set of the SMIv2 specification. Due to the added complexity this variant is not commonly implemented.

NETWORK MANAGEMENT
INMF: Remote Monitoring MIB

- The main goals of the RMON MIB is **to obtain traffic statistics from all network layers in a LAN domain segment.**
- The SNMP agents that implement the RMON MIBs should be special hosts with special access and security permissions to scan/collect all needed network traffic from all logical protocol layers from all network cards on the LAN. These are known as **RMON probes**.
- The RMON probes should use **passive monitoring** of network traffic.
- There should be, at least, one RMON probe for each LAN on the network where traffic monitoring is desired.
- Each RMON probe gathers/scans network traffic at a configurable rate and calculates statistics. Normally these are long term that are then polled by the managers for further analysis. In case of special conditions (like errors) the probe can trigger alarms/notifications.

NETWORK MANAGEMENT
INMF: Remote Monitoring MIB

- A RMON probe will scan traffic data at the data link layer but, if version 2 is supported (RMONv2), it can calculate statistics about all network layers (and even about application communication protocols).
- RMONv2 supports, when compared to RMONv1, additional matrix data statistics. RMONv1 focuses on traffic statistics at the data link layer only.
- For example, an RMON probe can calculate the following statistics (for its LAN domain segment):
  - ✓ Number of packets, bytes, broadcast/multicast packets, etc.
  - ✓ CRC errors, length problem, collisions, etc.
  - ✓ Size histograms
    [<64, 65-127, 128-255, 256-511, 512-1023, 1024-1518]

NETWORK MANAGEMENT
INMF: Remote Monitoring MIB

Excerpt from RMON MIB (history/sampling configuration):

```
historyControlEntry OBJECT-TYPE
    SYNTAX     HistoryControlEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "A list of parameters that set up a periodic sampling of
        statistics.  As an example, an instance of the
        historyControlInterval object might be named
        historyControlInterval.2"
    INDEX { historyControlIndex }
    ::= { historyControlTable 1 }

HistoryControlEntry ::= SEQUENCE {
    historyControlIndex              Integer32,
    historyControlDataSource         OBJECT IDENTIFIER,
    historyControlBucketsRequested   Integer32,
    historyControlBucketsGranted     Integer32,
    historyControlInterval           Integer32,
    historyControlOwner              OwnerString,
    historyControlStatus             EntryStatus
}
```

NETWORK MANAGEMENT
INMF: Remote Monitoring MIB

Excerpt from RMON MIB (ethernet statistics):

```
EtherStatsEntry ::= SEQUENCE {
    etherStatsIndex                  Integer32,
    etherStatsDataSource             OBJECT IDENTIFIEI
    etherStatsDropEvents             Counter32,
    etherStatsOctets                 Counter32,
    etherStatsPkts                   Counter32,
    etherStatsBroadcastPkts          Counter32,
    etherStatsMulticastPkts          Counter32,
    etherStatsCRCAlignErrors         Counter32,
    etherStatsUndersizePkts          Counter32,
    etherStatsOversizePkts           Counter32,
    etherStatsFragments              Counter32,
    etherStatsJabbers                Counter32,
    etherStatsCollisions             Counter32,
    etherStatsPkts64Octets           Counter32,
    etherStatsPkts65to127Octets      Counter32,
    etherStatsPkts128to255Octets     Counter32,
    etherStatsPkts256to511Octets     Counter32,
    etherStatsPkts512to1023Octets    Counter32,
    etherStatsPkts1024to1518Octets   Counter32,
    etherStatsOwner                  OwnerString,
    etherStatsStatus                 EntryStatus
}
```

NETWORK MANAGEMENT
INMF: Remote Monitoring MIB

Excerpt from RMON MIB (alarm/event/log configuration):

```
AlarmEntry ::= SEQUENCE {
    alarmIndex                  Integer32,
    alarmInterval               Integer32,
    alarmVariable               OBJECT IDENTIFIE
    alarmSampleType             INTEGER,
    alarmValue                  Integer32,
    alarmStartupAlarm           INTEGER,
    alarmRisingThreshold        Integer32,
    alarmFallingThreshold       Integer32,
    alarmRisingEventIndex       Integer32,
    alarmFallingEventIndex      Integer32,
    alarmOwner                  OwnerString,
    alarmStatus                 EntryStatus
}
```

```
EventEntry ::= SEQUENCE {
    eventIndex          Integer32,
    eventDescription    DisplayString,
    eventType           INTEGER,
    eventCommunity      OCTET STRING,
    eventLastTimeSent   TimeTicks,
    eventOwner          OwnerString,
    eventStatus         EntryStatus
}
```

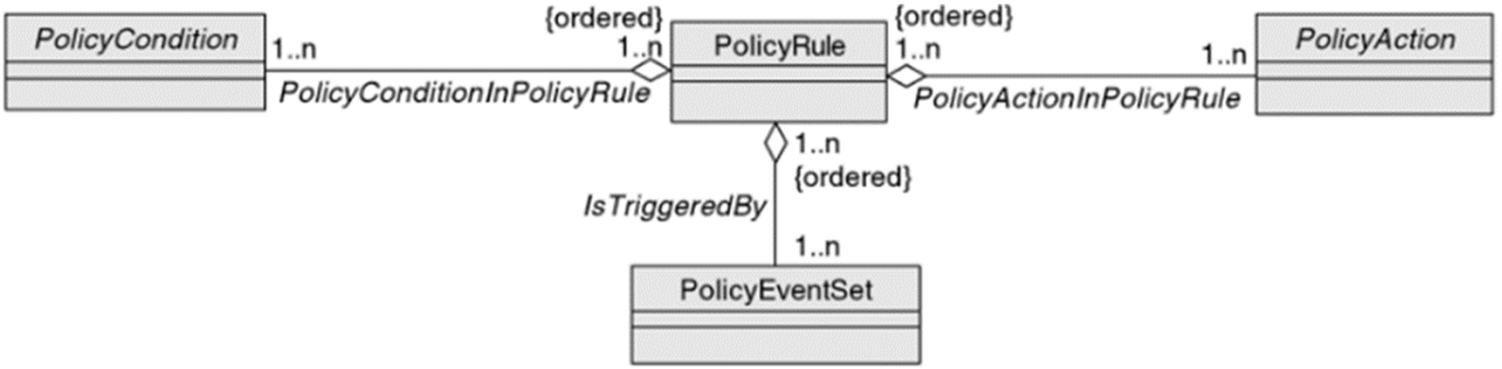
```
LogEntry ::= SEQUENCE {
    logEventIndex       Integer32,
    logIndex            Integer32,
    logTime             TimeTicks,
    logDescription      DisplayString
}
```

NETWORK MANAGEMENT
Policy-Based Network Management

- Simplification of network management hiding low level management procedures using a standard/universal information model. It's a smart and more efficient alternative to overprovisioning of resources.
- Models management behavior, not policy implementation details.
- It's complex to implement efficiently just by using traditional architectures like the INMF/SNMP.
- Supports less well trained network administrators or system managers, permits higher-level management functionalities and directly supports automation.
- It allows higher security levels by limiting functionalities per user groups and by continuously analyzing management results.
- Supports real-time and time-critical management procedures.

NETWORK MANAGEMENT
Policy-Based Network Management

- Management policies are defined by a set of rules.
- Each rule includes conditions and actions.
- If a condition is met then a pre-determined action is executed.
- Events can be defined for automatic verification of conditions.



NETWORK MANAGEMENT
Policy-Based Network Management

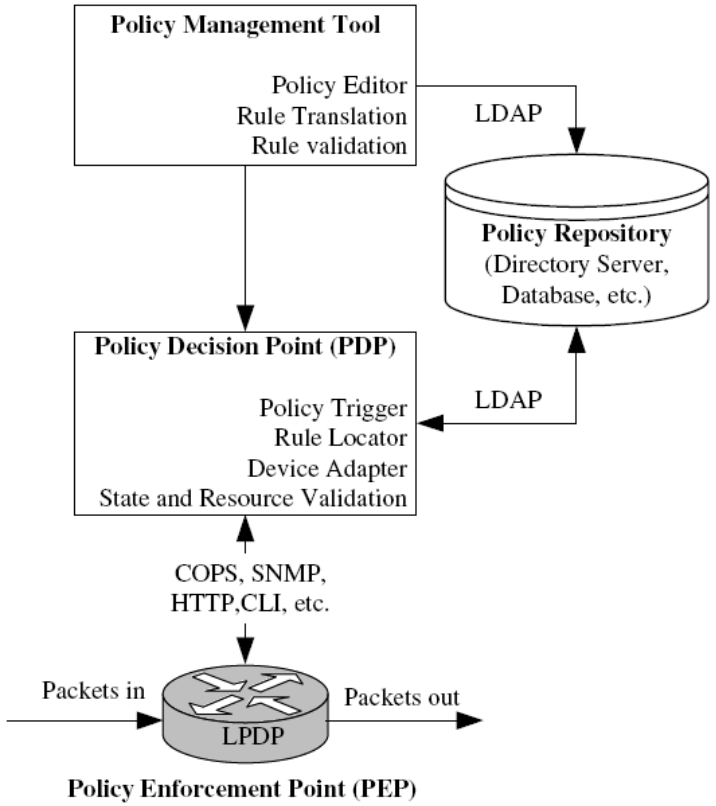
IETF RFC 3198

A standardization effort for policy management in the Internet. It integrates the following components:

- **Policy Decision Point** (PDP) – entity that manages/calculates policies for itself and to other entities in the management domain.
- **Policy Enforcement Point** (PEP) – entity that applies policies.
- **Policy Management Tool** – entity where policies can be edited, validated and converted.
- **Policy Information Repository** – entity that stores policy information (or PIB, like a MIB for policies).

NETWORK MANAGEMENT
Policy-Based Network Management

IETF RFC 3198



NETWORK MANAGEMENT
Policy-Based Network Management

IETF RFC 3198

A standardization effort for policy management in the Internet.
It includes the following standards:

- Common Open Policy Service
  (COPS, RFC 2748)
- Common Open Policy Service for Policy Provisioning
  (COPS-PR, RFC 3084 – used together with COPS)
- Structure of Policy Provisioning Information
  (SPPI, RFC 3159)
- Policy Information Bases
  (defined using the SPPI syntax)

NETWORK MANAGEMENT

Autonomic Network Management

A step further to automation…

- Supports building of automated systems to deal with higher management complexity.
- Reduces human/manual intervention (less personnel).
- Higher efficiency by constant feedback and faster readjustments of the management process.
- Behavior represented by management policies.
- It aims to implement a set of management self-functions: self-protecting, self-configuring, self-healing and self-optimizing.
- Supports Context-Awareness service management.

NETWORK MANAGEMENT

Autonomic Network Management

It has several key requirements:

- Policy-based management architectures and communication protocols that support automation, real-time and time-critical management procedures.
- Truly universal information models that can support the integration of any management service with any level of functionality.
- Management of network services are organized on layers of different abstraction, implementing different levels of management functionalities.

NETWORK MANAGEMENT
Network Services Management

Towards a full integration of management procedures…

FCAPS is no longer used to define management services and procedures, which can be divided into three main groups of services/procedures:

- **Operational Services Management**
  - Monitoring
  - Configuration
- **Administrative Services Management**
- **Strategic Services Management**

NETWORK MANAGEMENT
Network Services Management

Operational Management:

- Monitoring – Faults, Configuration, Quality of Services, Security & Accounting, etc.
- Configuration – Adaptation of strategic policies into administrative policies; Installation of all hardware devices and cabling; Installation and setup of all software needed; Configuration of devices and software services; etc.

NETWORK MANAGEMENT
Network Services Management

Administrative Management:

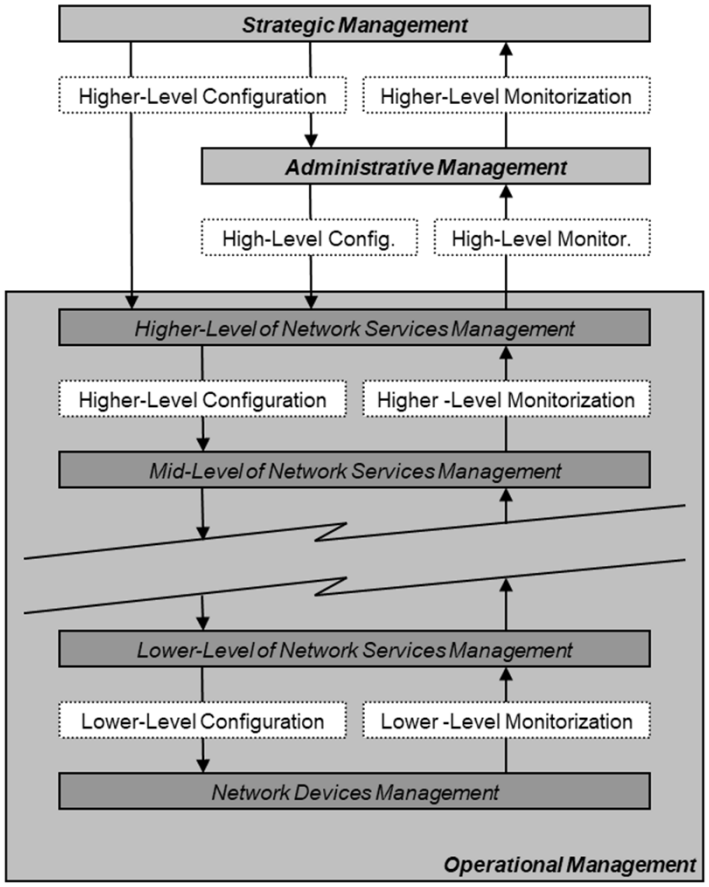
- Administrative assignment of addresses – Network addresses, domain names and interface addresses, etc.;
- Consumer and internal user support monitoring – Human live contacts, standardized consumer applications reports, snail-mail, internal communication documents, etc.;
- Consumer information data processing – Consumer reports or marketing documentation, etc.;
- Internal technical and informational reports – Input/output documents from/for operational and strategic management.

NETWORK MANAGEMENT
Network Services Management

Strategic Management:

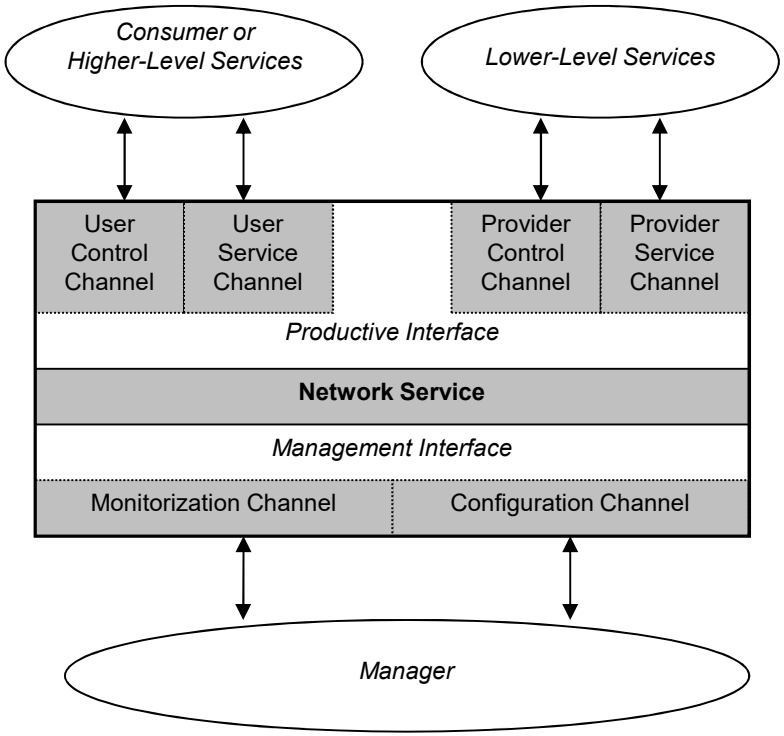
- Monitoring of administrative & operational reports – User satisfaction reports, operational management deployment and monitoring, etc.;
- Definition of services access policies – Security requirements, levels of quality of services and their pricing; naming & addressing strategies, etc.;
- Definition of a global consumer marketing and support policy;
- Coordination between all entities supporting the entire network operation, including network services management.

NETWORK MANAGEMENT
Network Services Management



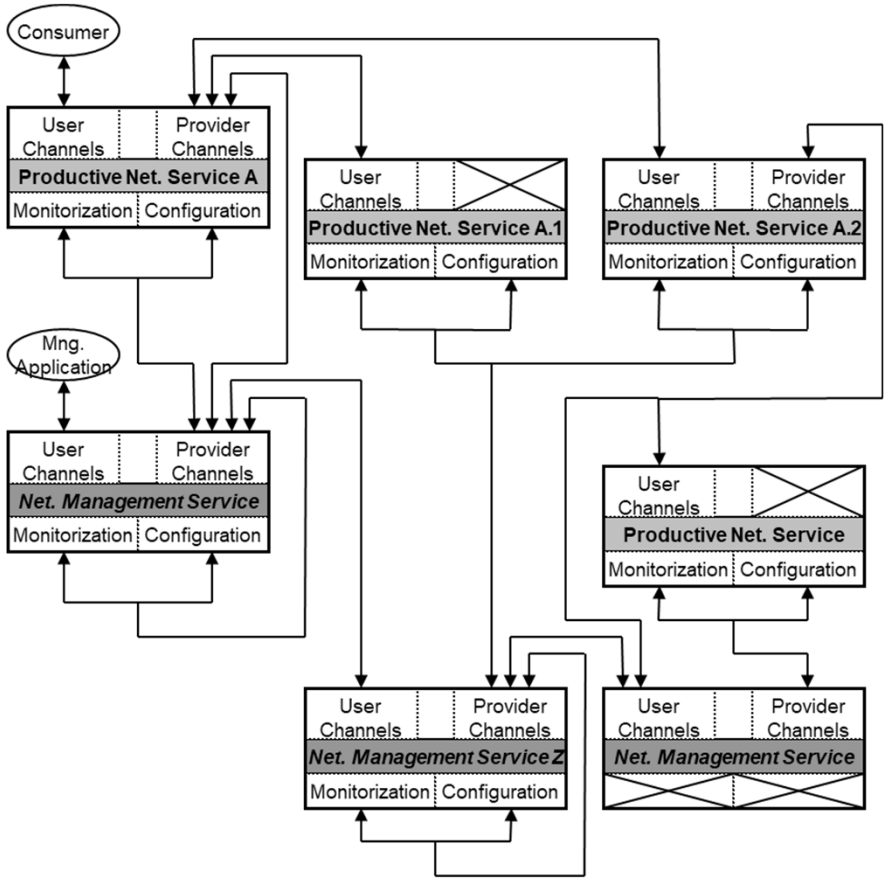
NETWORK MANAGEMENT
Network Services Management

Definition of a Network Service…



NETWORK MANAGEMENT
Network Services Management

An example…



NETWORK MANAGEMENT
Network Services Management

A framework for deployment of network services management should take into consideration the following components:

- **Architecture** – definition of the management entities, how they relate, the access and security policies, technologic constraints and administrative policies.
- **Communications Model** – definition of the services that compromise the entire management framework and the syntaxes and semantics of their user and management interfaces.
- **Functional Model*** – definition of the available Service Management Functions; in the case of management services, management procedures and their functionalities/parameters.

NETWORK MANAGEMENT
NETCONF

An alternative framework to SNMP created by Cisco Systems and that it was targeted for **configuration management** of routers. The key requirements (or selling points ;) were:

- Easy to use (at least easier than SNMP);
- Clear distinction between configuration data, operational state and statistics; support to multi-configuration data;
- Support for configuration of the entire network as a whole (master databases), rather than individual devices;
- Support configuration transactions across a number of devices;
- Support consistency checks of configurations;

NETWORK MANAGEMENT
NETCONF

An alternative framework to SNMP created by Cisco Systems and that it was targeted for **configuration management** of routers. The key requirements (or selling points ;) were:

- Support to role-based access control models and the principle of least privilege (à la Cisco Systems);
- Support for consistency and security checks of access control lists across devices in the network;
- Support to data-oriented but, more importantly, to task-oriented access control;
- Still maintain the same level of protocol simplicity and resource consumption of the SNMP framework.

NETWORK MANAGEMENT
NETCONF

The two essential components are:

- The NETCONF specification on RFCs 4741, 6241 and 8342 that defines a small set of operations, just like SNMP, but these are conveyed on a XML-based remote procedure call mechanism (encoded with UTF-8) that is task-oriented; it should be encapsulated over a connection-oriented transport service (usually SSH); all interactions between agents and managers should be carried out in NETCONF sessions; thus, security guarantees deployment is session-oriented.
- The **Yet Another Next Generation** (YANG) language for the specification of the configuration datastores (just like SMI for the SNMP framework), RFC 7950.

NETWORK MANAGEMENT
NETCONF

Operations defined on NETCONF RFC 4741 (later there was an update on RFC 6241 and another on RFC 8342):

Operation	Description
`<get>`	Retrieve running configuration and device state information
`<get-config>`	Retrieve all or part of specified configuration datastore
`<edit-config>`	Loads all or part of a configuration to the specified configuration datastore
`<copy-config>`	Replace an entire configuration datastore with another
`<delete-config>`	Delete a configuration datastore
`<commit>`	Copy candidate datastore to running datastore
`<lock> / <unlock>`	Lock or unlock the entire configuration datastore system
`<close-session>`	Graceful termination of NETCONF session
`<kill-session>`	Forced termination of NETCONF session