CHAPTER TWO, *About Network Management Frameworks & Technologies*

This page was intentionally left blank.

Acronyms & Terms

ACSE	2-15	Association Control Service Element
ASN.1	2-17	Abstract Syntax Notation 1
BER	2-17	Basic Encoding Rules
CCITT	2-13	Consultative Committee for International Telephone and Telegraph
CIM	2-32	Common Information Model
CMIP	2-13	Common Management Information Protocol
CMIS	2-13	Common Management Information Service
CMISE	2-15	CMIS Element
CMOT	2-17	CMIP Over TCP/IP
COPS	2-32	Common Open Policy Service
CORBA	2-33	Common Object Request Broker Architecture
DISMAN	2-25	IETF Distributed Management Working Group
DMTF	2-29	Distributed Management Task Force
EACM	2-11	Entity Access Control Model
FTP	2-26	File Transfer Protocol
HEMS	2-17	High-Level Entity Management System
HTTP	2-26	Hypertext Transfer Protocol
IAB	2-17	Internet Activities Board
IMA	2-30	Intelligent Mobile Agents
INMF	2-12	Internet-standard Network Management Framework
IRTF	2-27	Internet Research Task Force
ISO	2-13	International Organization for Standardization
M2M	2-25	Manager-to-Manager
MDM	2-12	Management Data Model
MDRIS	2-11	Management Domain Registration and Information Service
MIB	2-15	Management Information Base
MIT	2-17	Management Information Tree
NMRG	2-27	Network Management Research Group
NMSAP	2-11	Network Management Service Access Point
NSMF	2-5	Network Services Management Framework
OID	2-24	Object Identification
OMG	2-12	Object Management Group
OSI	2-12	Open Systems Interconnection
OSI/NMF	2-13	OSI Network Management Framework
PDL	2-32	Policy Description Language
PDU	2-22	Protocol Data Unit
PIB	2-27	Policy Information Base
RMON	2-24	Remote Monitoring MIB
SBSM	2-24	Session-Based Security Model
SGMP	2-17	Simple Gateway Management Protocol
SMI	2-13	Structure of Management Information
SMIng	2-27	SMI for the Next Generation
SNMP	2-15	Simple Network Management Protocol
SNMPConf	2-27	IETF Configuration Management with SNMP Working Group
SPPI	2-27	Structure of Policy Provisioning Information
TCP	2-23	Transmission Control Protocol
TMN	2-13	Telecommunications Management Network
UDP	2-23	User Datagram Protocol
UIM	2-12	Universal Information Model
USM	2-22	User-based Security Model
VACM	2-22	View-based Access Control Model
WEBM	2-33	Web-Based Enterprise Management framework

List of Figures

List of Tables

1 Introduction

This chapter will be a journey into the Computer Networks Management research field, providing an overview of what the author feels it has been relevant in this field for the past decade. It should be noted that it is not a neutral presentation because the author will take the opportunity for presenting the eventual merits and limitations of the presented technologies based on his personal knowledge and research, sometimes not in accordance with others visions and opinions that can be revisited through the referenced material listed on the References section ending this thesis.

Furthermore, this chapter will be presented in a way that will also serve to justify all the architecture, major concepts, technologies and mechanisms created for or used in the Network Services Management Framework (NSMF).

The rest of this section will introduce major concepts of modern network services management and management information models. These concepts will be fundamental for a better understanding of many network management technologies, and their evaluation, presented throughout this chapter. Section 2 and Section 3 will present a summary of the most important concepts and mechanisms, their merits and limitations, behind the two most influential and deployed management frameworks. The last section of this chapter will present a survey of the most important trends on this research field and how they affected the creation of the NSMF.

1.1 Definition of Network Management

There are many definitions for what could be considered as *network management*; none seems to be final and all seem to share common concepts. While some definitions tend to favor an immediate systematic approach with enumeration of the network management goals, this section will start to present an overall heurist definition already introduced on [30] and them will provide a complementary and systematic list of its main objectives.

Network Management is the set of activities that enables a network to meet the operational (or functional), administrative and strategic goals set by the network administration. These goals will take into consideration the available resources (both technical and economical), the consumer[i] needs (humans or applications that make use of the network) and the technical, economical and marketing goals of the humans controlling the management process (from network operators to the administration board – or similar – of the organization).

Taking a more systematic approach, we can further define these three groups of activities, without being too detailed on explanations about well known concepts associated with network management:

- Operational Management – this type of management deals with all the functional aspects of the network operation from the network hardware and network services installation and configuration to fault monitorization and recovery procedures; moreover, this includes:
    - Monitorization Management of all network services operations at all layers and abstraction levels, which includes:
        - fault monitorization (detection of operational faults on the network and the on network services through examination of monitoring statistics or consumer reports),
        - configuration monitorization (detection of configuration errors due to abnormal functioning of network devices or services software, due to incorrect configuration procedures by network operators or in consequence of a malicious access),
        - quality of services monitorization (which includes monitorization of performance, service levels, global resource consumption, etc),
        - security monitorization (consumer and any type of user access monitoring and logging, intrusion detection, incorrect network services usage, etc), and

[i] The author prefers the use of this term in this context because it denotes only the end user of the network services and not the user of other internal services, like the network management service. Each consumer is represented by a service level agreement, which may involve one individual or an organization and associated end-user applications.

        - accounting monitorization (traffic classification and accounting, consumer traffic and time accounting, etc); it should be noted that monitorization only deals with the detection, registration, logging and report of these situations previously described; it does not involve any active changing of operational or policy configurations of network services and network devices;
    - Configuration Management of all network devices and services at all layers and abstraction levels, which includes:
        - Adaptation of strategic policies into administrative policies and requisites for all aspects of operational management, at any layer or level of abstraction (typically, the configuration management will be deployed as independent management procedures using different types of enabling technologies at different layers or abstraction levels),
        - Installation of all hardware devices and cabling needed to deploy the planned network; also, installation and setup of all software needed to initiate the planned network services, from the link layer up to the application layer, including a network management system,
        - Re-active configuration of devices and software services in result of monitorization reports, changes in administrative or strategic policies, etc;
- Administrative Management – this type of management deals with all administrative procedures needed to assure that all technical requisites can be met and that follow all strategic requirements; for example:
    - When applied, administrative assignment of network addresses, domain names and interface addresses;
    - Consumer and internal user support monitorization, which may involve several types of interface with users, some of them external to the deployed network services (human live contacts, standardized consumer applications reports, snail-mail, internal communication documents, etc);
    - Consumer information data processing (as defined by the strategic access policies) and creation and publication of consumer technical reports or marketing documentation;
    - Creation and publication of technical and informational reports to be used as input documents for operational and strategic management;
    - Creation and publication of internal network services operation cost reports and individual consumer account reports;
- Strategic Management – finally, this type of management defines the global policies that better meet the network administration goals; these policies, which must be transformed into administrative and operational requirements include:
    - Monitorization of administrative informational and accounting reports about consumer usage, level of satisfaction, etc;
    - Monitorization of administrative technical reports about all aspects of operational management deployment;
    - Definition of network services access policies for the consumers (or network service users) and for the internal network services (including the management service);
    - Definition of global security requirements to be deployed on the network; The strategic security policies will define several levels of security deployment, depending on the service being offered and the consumer using it, or the internal service being managed and the user managing it;
    - Definition of levels of quality of services/costs policies;
    - Definition of global naming and addressing strategies for all services;
    - Definition of a global consumer marketing and consumer support policy;
    - Definition and coordination between all entities supporting the entire network operation, including network services management;
    - Re-definition of global policies based on the pure shift of strategic goals or as a result of analysis of administrative reports.

The relationship between all these aspects of network management can be better understood with the complementing Figure 1. It can be seen that every type of management is linked through two types of interaction or management procedures: monitorization, which can be done using two known strategies (polling or surveys, and notifications or reports),[i] or configuration. Both management procedures can be

[i] These can be applied individually or combined, which is most probable.

done at any layer of the management process and have any level of functionality; furthermore, they can include associated complex manipulation and processing techniques of management data.

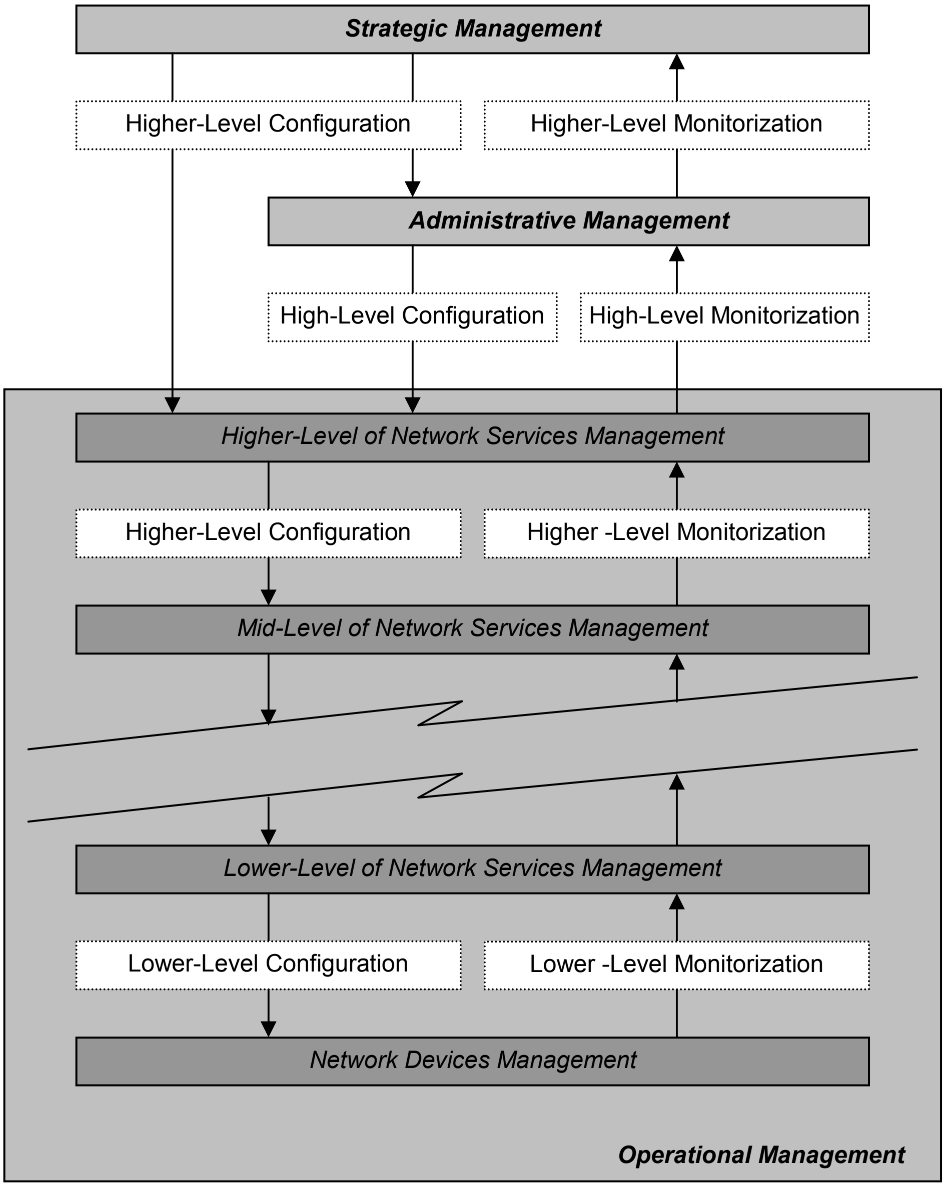
Strategic Management
Higher-Level Configuration
Higher-Level Monitorization
Administrative Management
High-Level Configuration
High-Level Monitorization
Higher-Level of Network Services Management
Higher-Level Configuration
Higher -Level Monitorization
Mid-Level of Network Services Management
Lower-Level of Network Services Management
Lower-Level Configuration
Lower -Level Monitorization
Network Devices Management
Operational Management

Figure 1: Generic structure for Network Management.

The lower management level depicted can include itself various functionality levels of network management, but it is desirable that very lower levels of device management be hidden to the network management system by its automatic inclusion on the operating system of the network devices. Each network services management level is formed by the management service of various network services at that layer or functionality level.

In general, the activities of strategic and administrative management are carried out by humans because their requirements are generic and heuristic. The major concern for network management frameworks is to provide a set of concepts, standards and technologies for an effective deployment of operational management, although aspects of strategic and administrative management can be scientifically approached and studied.

1.2 Network Services Management versus Network Management

When the first network management frameworks appeared they were concentrated on the management of the network devices that formed the network and were designated generically as frameworks for network management. Today, this notion is dated and networks are service oriented with the service providers

trying to hide the network technical details from the consumers (or users of their network/applications services) and establishing service level agreements.
This concept of network service can be transposed to internal deployment of network production protocols, mechanisms, utilities or applications.[i] So, each set of related activities that is executed, on an individual or on a group of network devices, to provide the deployment of the same goal can be grouped under the concept of a network service.[ii] Further interesting readings on this subject can be found on [92], although with restricted applicability to the Internet.

The advantage of this representation is that each network service can be modelled as a black box implementing a set of deployment requisites with any required level of functionality and independently of any specific device or technology. This box, depicted on Figure 2, will be completely defined by its interfaces that both its implementors and its users must comply to. Conceptually, each network service box must define its productive service interface (the service that its users will expect to be provided with and the lower-level services that the service itself expects to be available on the network) and its management service interface (the service available for its management). Both types of service interface are further divided on four and two interface channels, respectively: two pairs of control and service channels for the productive interface and a pair of monitorization and configuration channels for the management interface. The interaction between all network services is made through these interface channels.

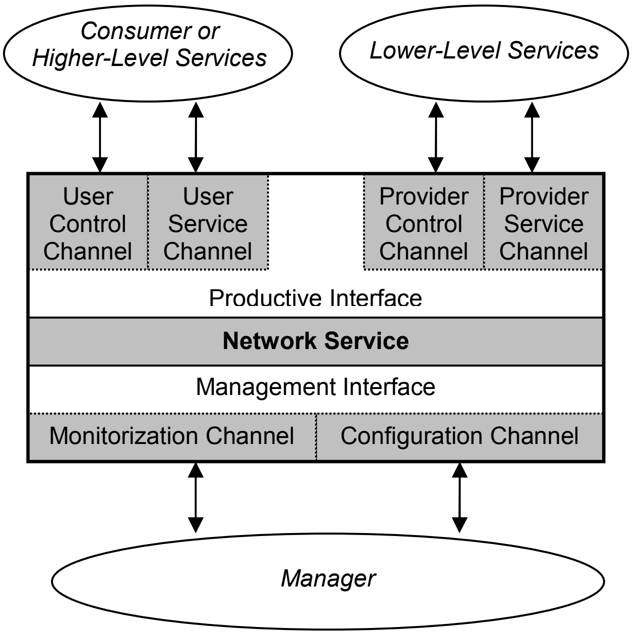

Figure 2: Conceptualization of a Network Service.

The user control channel of the productive interface serves for the user to control its service channel (for example, it can be used for admission of access control information from the user, for requesting specific levels of quality of service, for requesting user support, etc). The user service channel is for exchange of user data or requesting the execution of service procedures defined for each network service (transmission of an email message, establishment of a multimedia connection, mounting a remote file system, obtaining a web page, mapping a network address into a physical address, etc). The provider control and service channels have analogous utilizations but for interaction with lower-level network services. In this case, the network service becomes the user of these lower-levels network services. The monitorization and configuration channels of the management interface can be used to request the execution and to obtain results of management activities related to monitorization and configuration management, as defined on the previous section.

[i] The network deployment details are internal if they are only relevant to the network administration and operators or managers and not to the consumer.
[ii] This notion of network service can be extended to strategic and administrative management. The activities of these two types of network management can be abstracted as a set of network services, even if they relate to management activities that are human oriented.

Each network service is completely defined by defining its interface channels. Each network service will implement the desired functionality level and will define the interface channels using the appropriate technologies. Even the network devices at the physical and link level can be modelled as network services. Each end-user (or consumer) network service is implemented as one hierarchic tree of network services, with each level adding extra functionalities and abstraction levels. The mid-level network services will have other higher-level network services as its users and will be users of other lower-level network services.

So, in this thesis, the term network services management will have the same semantics as the term network management, since the overall management of a network of services is obtained with the management of all its services. Furthermore, the management activities are also modelled as a set of network services, in this case management services.

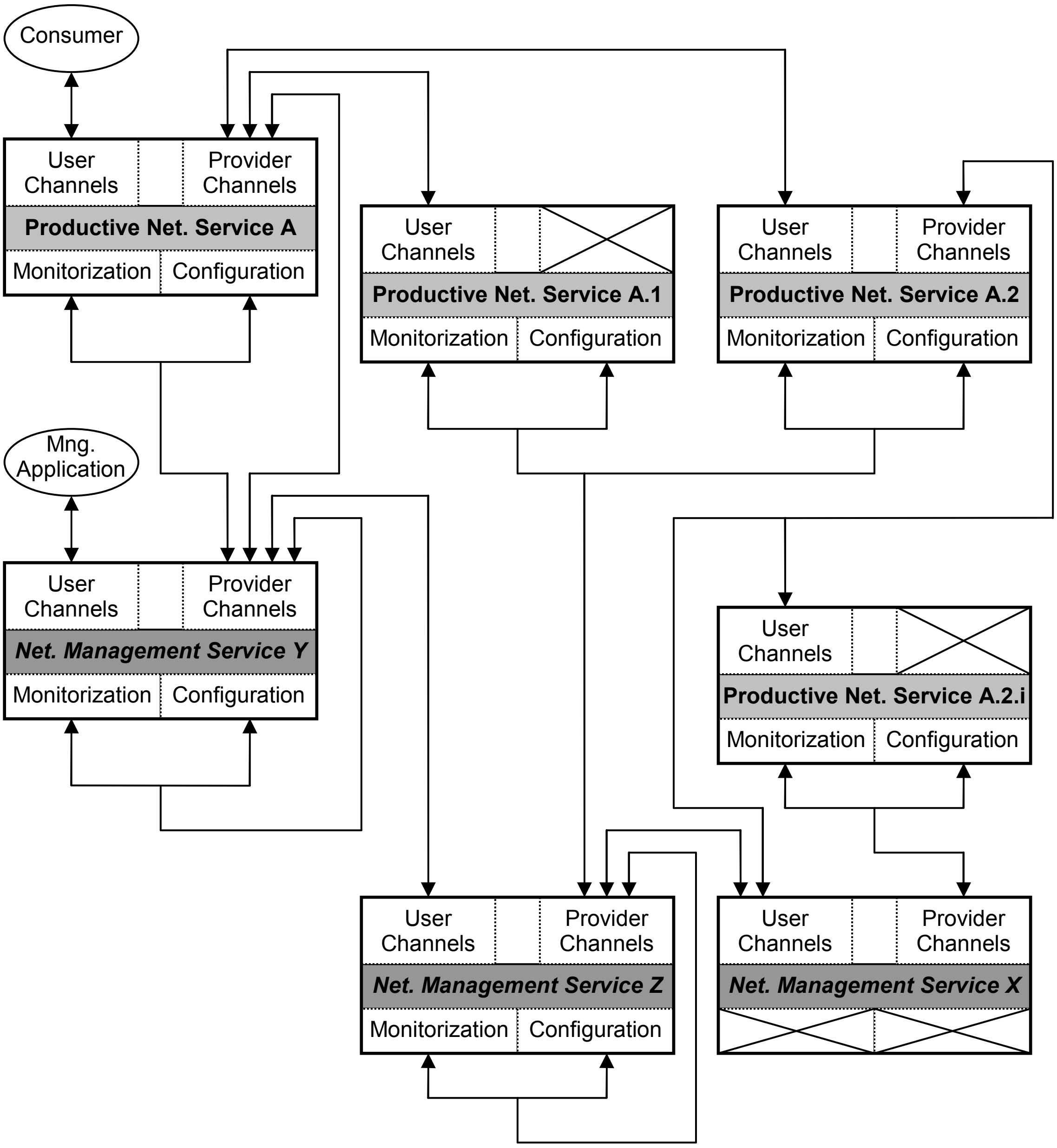

Figure 3: Example of a network services system with its productive and its management services.

The entire network can be modelled as a graph of network services, and, more importantly, its management can be represented as a graph of special management services that will manage the productive management services.[i] These primary network management services[ii] will also be managed by secondary network management services and so on. In general, and because device and network resources are not unlimited, only the primary network management services are defined and implemented, which in this case, should include procedures for its own management.

Figure 3 summarizes the concepts of network services and network services management just presented and that were the basis for the definition of the Network Services Management Framework. In the example, the users of the productive network service A are end consumers, unlike the other productive services of the figure, that don't provide lower-level services to the higher-level network service. The productive services A.1 and A2.i are leaf network services, probably for device management, and don't expect to use any other services for its implementation.

Services Y, Z and X are management services and service Y is the higher-level management service and will interact with a management application. The other network management services will provide lower-levels management services to this service. It should be noted that this network platform uses a single network services management dimension, since the management services are all managed by itself or, in the case of the of service X, the network management service itself can not be managed externally.

Another important feature is that productive network services can actively use the network management services, which normally is not obvious or directly contemplated on other approaches.

Finally, it becomes obvious the hierarchical nature of this organization of network services, including network management services. All these concepts are adopted by the NSMF which adds the notion of management domain: a group of network services related by a set of common management goals with the same global management policy. This trend of network services management, in opposition to traditional network management, is also debated on the Cloud to Cloud framework [136], although the modelling of the network services interfaces is less detailed are the framework is tied to an information model.

1.3 Management Domains & Management Entities

We have described the structure of the network services management framework and how the several services integrate and interact. Since this conceptualization permits hierarchies of management services with distinct management goals, it may be useful to create different network services for management of the same technologies but associated with different functionality levels (or abstraction levels). Or, it could be useful to group a set of network services administratively, considering a common administrative setup (for example, with common user access policies and security policies, and sharing the same network administration).

These considerations lead to the notion of a hierarchic structure of Management Domains. Each management domain groups a set of network services that share common management domain policies and network administration. This type of distributed organization is adopted by the NSMF and the concept itself, although not directly applied to this type of network management service definition, was defined for some time now, with good examples on [17,25,168]. Each management domain should be identified by a Management Services Identification, which in the case of the NSMF is a Management Domain Name.[iii]

A network service must be implemented by some sort of software executing on a group of network devices (although some lower-level services could be implemented just on hardware). The notion of an atomic group of resources implementing a network service with a sole responsible administration is

[i] The productive management services are the ultimate goal of the network infrastructure since they are the services to be provided to the consumers. The network services that directly manage these productive services are named first dimension or primary network management services.

[ii] Second dimension or secondary network management services only manage primary network management services and no productive network services. The third dimension network management services only manage the secondary network management services and so on. Additionally, each network management service can manage itself or other management services on the same management dimension.

[iii] Each NSMF Domain Name is a concatenation of sub-names, representing the hierarchic structure of the domains.

embossed on the concept of a Services Entity, and in the case of management services, a Management Services Entity or just Management Entity, and their identification should be obtained using a Management Entity Identification. In the NSMF, this management service entity identification is an Entity Name, which is formed by the management domain name prefixed with a unique entity name inside the domain.

The productive and management interfaces of a network service must be identified by a Network Service Access Point that, in the case of a network management service, will be a Network Management Service Access Point (NMSAP).[i] The syntax and semantics associated with a NMSAP should contain information on the syntax and semantics rules on how to exchange management data, either management procedures execution requests or management procedures results.

Since, the network services are implemented on network services entities some method of association must be provided for mapping entities identifications and services access points, or in the case of the NSMF, mapping management entities names into NSMAPs, which is defined on the Entity Access Control Model (EACM). Furthermore, the Management Domain Registration and Information Service (MDRIS) is the component of the EACM responsible for implementation of this mapping functionality, as explained on Chapter Four.

1.4 Components of a Network Services Management Framework

While the analysis of the previous sections lead to the introduction of important concepts for network management (these were: network services, management domains and management entities) and for the NSMF in particular, this could be completed by an analysis on the informational aspect of network management. This is also relevant because many, if not all, major network management frameworks are information oriented (not to be confused with object-oriented) and their definition is overly dependent on some specific information model and/or management data model.[ii]

As it can be implied from what was presented earlier, and following our network services and network management services conceptualization, a network services management framework is defined by:

- An Architecture – all management entities, how they relate (management domains, for example), the access and security policies, technologic constraints and administrative policies to be complied by entities implementing those network management services;
- A Communications Model – all network management services that compromise the entire management framework and the syntaxes and semantics of their user and management interfaces;[iii] and
- A Functional Model – the available service functions, in the case of management services, management procedures and their functionalities (that is, for each network management service, the set of expected management procedures, their parameterization and possible results) of each network management service.[iv]

Although these components seem enough to the author of this thesis, the true is that almost all management frameworks add another level of compliance, even for frameworks that intend to be generic, which, in this author's opinion, should be left to be defined by specific network management services platforms and not to the definition of management frameworks. This extra component is an Information Model. So, a network services management framework must be defined by the previous three components, and, optionally, by four components, with the inclusion of an Information Model, which complies with the definition found on [136], although in this case, the Information Model is mandatory.

[i] While not commonly used, a NMSAP can be divided on several sub-identifiers, either one for each channel of the productive interface and of the management interface, or one for each interface. As will be seeing on Chapter 4, the NSMF uses one identifier for all these channels. Furthermore, each network services management domain will define a group of supported NMSAP syntaxes and semantics.

[ii] Which, as it will be seen, is not the case of the NSMF.

[iii] This may include a mapping service for domain and entities names, a security key management service, etc.

[iv] All mechanisms or technologies to be implemented by the framework should be defined as service functions and must be used/accessed through the standard services interfaces.

Management Information Model & Management Data Models

The need for an information model for a network service is understandable (although not imperative, and, sometimes, it becomes a limiting factor on the functionality level and flexibility of the service) since it is a way of limiting the functionality provided by the service, which tends to make its implementation simpler but with the cost of an added complexity on its usage when higher-level service functions are needed. The solution for this limitation is to define a layered information model, where objects on higher layers have increased functional or abstraction levels (object-oriented technologies can be used) and lower layers will define objects with low-level management functionalities. A good and generic classification using this type of approach is made on [133,149], identifying three layers for management information models:

- An higher layer comprised by an Universal Information Model (UIM) that contains management objects that should be independent of the network management services technologies and, thus, of the management architecture or framework; this layer can be further divided on a higher level Conceptualization UIM and a lower level Specification UIM for an easier two step derivation of the UIM into the management data model;
- A middle layer comprised by a Management Data Model (MDM) that is dependent of the chosen network management architecture or adopted management framework but, if possible, should be independent on the representation, encoding and transmission technologies;
- A lower layer comprising the underlying representation, encoding (or codification) and transmission technologies used by the network management services.

Figure 4 depicts the tree entailed by this classification scheme and will be used later on with specific examples for each layer.

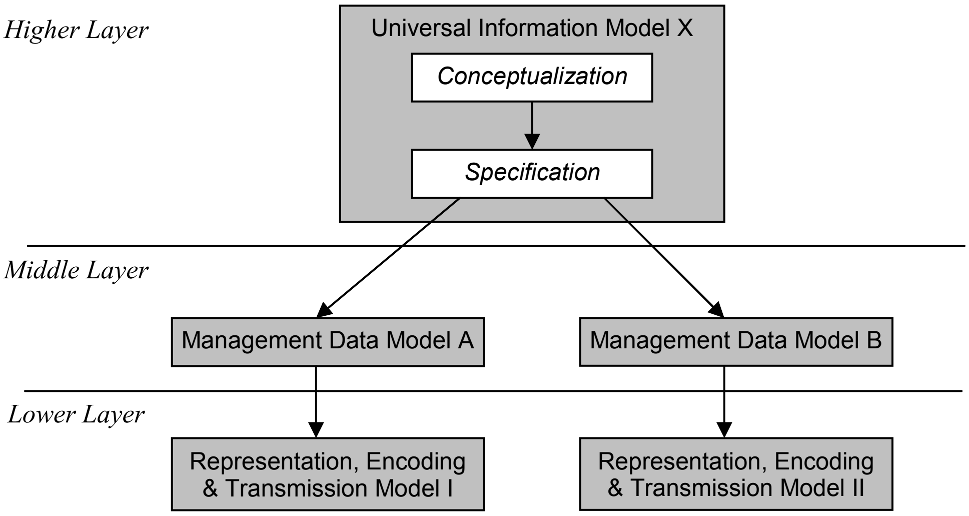
Higher Layer
Universal Information Model X
Conceptualization
Specification
Middle Layer
Management Data Model A
Management Data Model B
Lower Layer
Representation, Encoding & Transmission Model I
Representation, Encoding & Transmission Model II

Figure 4: An information architecture for network management.

Many state-of-the-art network management technologies make use of this entire information model (its major costumers are technologies based on Policy Management) while other more standard frameworks use only some of the layers, like the Internet-standard Network Management Framework (INMF) or the Open Systems Interconnection (OSI) approaches.

Other Information Models have been proposed, but the work described on [154] should be pointed out. This complex approach tries to integrate several existent information models into the same information model framework that it is inspired on the four layer information model defined by the Object Management Group (OMG) [170].

More esoteric work can be found on [18] and [140]. These trends take an extremely theoretic view of management information models. The research on [140] is restricted to fault management and it is very complex, which limits its current relevance, but the older work on [18] tried to define the deployment of formal methods to network management, which can be of great relevance for definition of the higher levels of an information model for network services management.

To end this section it should be remembered that the NSMF is not tied to any information model; its definition relies only on the three components defined on the beginning of this section: architecture, communications model and functional model.

2 The OSI Network Management Framework

This was the first real open management framework to be defined in the late 80's and first years of the 90's. While its direct deployment has not been widely successful on computer networks, it has influenced major network management frameworks like the INMF. At the same time the OSI framework was developed by the International Organization for Standardization (ISO) the Telecommunications Management Network (TMN) was also developed by the Consultative Committee for International Telephone and Telegraph (CCITT) for application on telecommunications networks.

Although the OSI Network Management Framework (OSI/NMF) is defined on many ISO documents, the most important are the ones that define the Common Management Information Service (CMIS) [2], the Common Management Information Protocol (CMIP) [7] and the Information Model based on its Structure of Management Information (SMI) [16]. For completeness, Table 1 presents the list of all OSI management standards.

ISO 7498-4: "Information Processing Systems - Open Systems Interconnection - Basic Reference Model - Part 4: Management Framework", Geneva, 1989.
ISO 9595: "Information Processing Systems - Open Systems Interconnection - Common Management Information Service Definition", Geneva, 1990.
ISO 9595/DAM 1: "Information Processing Systems - Open Systems Interconnection - Common Management Information Service Definition - Amendment 1: Cancel / Get", Geneva.
ISO 9595/DAM 2: "Information Processing Systems - Open Systems Interconnection - Common Management Information Service Definition - Amendment 2: Add, Remove and setToDefault", Geneva.
ISO 9595/PDAM 3: "Information Processing Systems - Open Systems Interconnection - Common Management Information Service Definition - Amendment 3: Support for Allomorphism", Geneva.
ISO 9595/DAM 4: "Information Processing Systems - Open Systems Interconnection - Common Management Information Service Definition - Amendment 4: Access Control", Geneva.
ISO 9596: "Information Processing Systems - Open Systems Interconnection - Common Management Information Protocol", Geneva, 1991.
ISO 9596/DAM 1: "Information Processing Systems - Open Systems Interconnection - Common Management Information Protocol - Amendment 1: Cancel / Get", Geneva.
ISO 9596/DAM 2: "Information Processing Systems - Open Systems Interconnection - Common Management Information Protocol - Amendment 2: Add, Remove and setToDefault", Geneva.
ISO 9596/PDAM 3: "Information Processing Systems - Open Systems Interconnection - Common Management Information Protocol - Amendment 3: Support for Allomorphism", Geneva.
ISO DIS 9596-2: "Information Processing Systems - Open Systems Interconnection - Common Management Information Protocol - Part 2: Protocol Implementation Conformance Statement (PICS) proforma", Geneva.
ISO 10040: "Information Processing Systems - Open Systems Interconnection - Systems Management Overview", Geneva.
ISO DIS 10164-1: "Information Processing Systems - Open Systems Interconnection - Systems Management - Part 1: Object Management Function", Geneva.
ISO DIS 10164-2: "Information Processing Systems - Open Systems Interconnection - Systems Management - Part 2: State Management Function", Geneva.
ISO DIS 10164-3: "Information Processing Systems - Open Systems Interconnection - Systems Management - Part 3: Attributes for Representing Relationships", Geneva.
ISO DIS 10164-4: "Information Processing Systems - Open Systems Interconnection - Systems Management - Part 4: Alarm Reporting Function", Geneva.
ISO DIS 10164-5: "Information Processing Systems - Open Systems Interconnection - Systems Management - Part 5: Event Report Management Function", Geneva.
ISO DIS 10164-6: "Information Processing Systems - Open Systems Interconnection - Systems Management - Part 6: Log Control Function", Geneva.

Table continues on next page…

Table 1: ISO/OSI Management Standards.

Table continuation…

ISO 10164-7: "Information Processing Systems - Open Systems Interconnection - Systems Management - Part 7: Security Alarm Reporting Function", Geneva.
ISO DIS 10164-8: "Information Processing Systems - Open Systems Interconnection - Systems Management - Part 8: Security Audit Trail Function", Geneva.
ISO CD 10164-9: "Information Processing Systems - Open Systems Interconnection - Systems Management - Part 9: Objects and Attributes for Access Control", Geneva.
ISO DIS 10164-10: "Information Processing Systems - Open Systems Interconnection - Systems Management - Part 10: Accounting Meter Function", Geneva.
ISO DIS 10164-11: "Information Processing Systems - Open Systems Interconnection - Systems Management - Part 11: Workload Monitoring Function", Geneva.
ISO DIS 10164-12: "Information Processing Systems - Open Systems Interconnection - Systems Management - Part 12: Test Management Function", Geneva.
ISO CD 10164-13: "Information Processing Systems - Open Systems Interconnection - Systems Management - Part 13: Measurement Summarization Function", Geneva.
ISO CD 10164-sm: "Information Processing Systems - Open Systems Interconnection - Systems Management - Part sm: Software Management Function", Geneva.
ISO CD 10164-tc: "Information Processing Systems - Open Systems Interconnection - Systems Management - Part tc: Confidence and Diagnostic Test classes", Geneva.
ISO CD 10164-ti: "Information Processing Systems - Open Systems Interconnection - Systems Management - Part ti: Time Management Function", Geneva.
ISO DIS 10165-1: "Information Processing Systems - Open Systems Interconnection - Structure of Management Information - Part 1: Management Information Model", Geneva.
ISO DIS 10165-2: "Information Processing Systems - Open Systems Interconnection - Structure of Management Information - Part 2: Definition of Management Information", Geneva.
ISO DIS 10165-4: "Information Processing Systems - Open Systems Interconnection - Structure of Management Information - Part 4: Guidelines for the Definition of Managed Objects", Geneva.
ISO CD 10165-5: "Information Processing Systems - Open Systems Interconnection - Structure of Management Information - Part 5: Generic Management Information", Geneva.
ISO CD 10165-6: "Information Processing Systems - Open Systems Interconnection - Structure of Management Information - Part 6: Requirements and Guidelines for Management Information Conformance Statement Proformas", Geneva.
ISO 10733: "Information Processing Systems - Open Systems Interconnection - Specification of the elements of Management Information related to OSI Network layer Standards", Geneva.
ISO 10733/PDAM1: "Information Processing Systems - Open Systems Interconnection - Specification of the elements of Management Information related to OSI Network layer Standards - Amendment 1: Managed object conformance statement proforma", Geneva.
ISO 10737: "Information Processing Systems - Open Systems Interconnection - Specification of the elements of Management Information relating to OSI Transport layer Standards", Geneva.
ISO 10737/PDAM1: "Information Processing Systems - Open Systems Interconnection - Specification of the elements of Management Information relating to OSI Transport layer Standards - Amendment 1: Specification of the elements of management information relating to NCMS", Geneva.
ISO 10737/PDAM2: "Information Processing Systems - Open Systems Interconnection - Specification of the elements of Management Information relating to OSI Transport layer Standards - Amendment 2: Managed object conformance statement proforma", Geneva.
ISO 10742: "Information Processing Systems - Open Systems Interconnection - Elements of Management Information related to OSI Data Link layer Standards", Geneva.
ISO TR ???: "Information Processing Systems - Open Systems Interconnection - Systems Management Tutorial - 2nd draft", ISO/IEC - JTC 1/SC 21/WG /N 1532, May 1992.

Table 1 (continuation): ISO/OSI Management Standards.

The author would like to remember the work of the OSI Network Management Forum that brought much insight into the development of the OSI/NMF architecture and pushed its completion and deployment. One of its major contributions was an article [3] about the architecture and main concepts behind the OSI/NMF. This article was influenced by the standards available at the time and would influence some important future standards of this framework.

Many of the OSI/NMF architecture concepts and functionalities were also adopted by the TNM framework[i] and would shape the development of the emerging, at the time, Simple Network Management Protocol (SNMP) and other components of the INMF.

2.1 Architecture

Its architecture is very simple and based on the Agent/Manager paradigm but, as with the NSMF, this roles are meant to be applicable on a per management communication basis, that is, a network management system (or management application) can act in the role of an agent on some management communications. This conceptual difference, not always understood on this framework, could be of importance because it makes this framework less prone to scalability problems than other entity agent/manager based frameworks.

Figure 5 depicts the overall architecture of this framework, with the agent process and the manager process communicating through CMIP and using the CMIS service supported by the CMIS Element (CMISE). Since the management entities are OSI application processes the CMISE uses the standard Remote Operations Service Element (ROSE), the System Management Application Service Element (SMASE) and the Association Control Service Element (ACSE), from the application layer of the OSI framework. Additionally, it should be noted that the management entities must be supported by the implementation of the entire OSI protocol stack. The network services/resources to be managed must be abstracted or conceptualized on objects grouped in Management Information Bases (MIB).

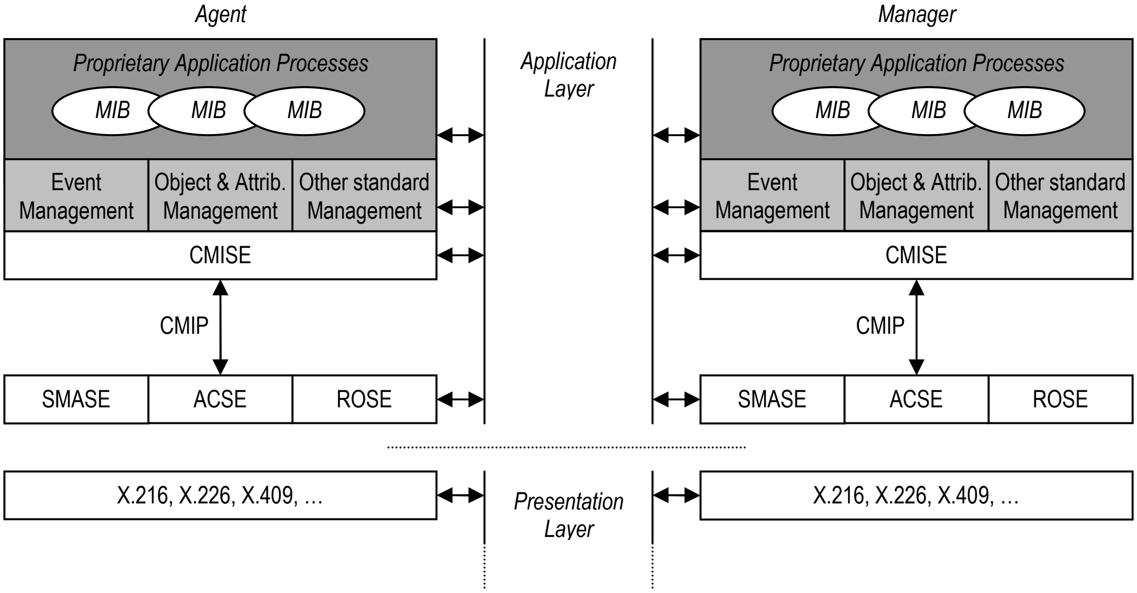

Figure 5: Architecture of the OSI/NM Framework.

As for security and access control deployment, the OSI/NMF defined that the management system should use the same security and access control technologies and services that were provided for all network services on the OSI stack.

2.2 Communications & Functional Models

The OSI/NM framework communications and functional models are tightly related and are based on the definition of the CMIS and its deployment method into the underlying OSI standard application services.

The functional components (like the event management and object and attributes management components) must use the same primitives applied to a selection of instances of objects of a MIB:[ii]

[i] Please consult [35] for a more detailed description on the TMN framework.

[ii] These primitives are defined in pairs, with one of them to be associated to the manager (primitive requests) and the other to the agent (primitive responses). Of course, its utility is also different, depending on the entity's management role.

M-Create, M-Delete, M-Get, M-Cancel-Get, M-Set, M-Action and M-Event-Report. These standard primitives are grouped on the event management and object & attributes management components of Figure 5. The M-Initialize, M-Terminate, and M-Abort are additional primitives for management communications services and are grouped on the other standard management component of the same figure.

Although not always recognized, the most important primitive should be the M-Action since it gives some flexibility to the framework by allowing the execution of management procedures associated with a management object.

The CMISE deployment assumes a connection-oriented transport protocol, adding an extra reliability to the management communications. On the other hand, some management services or management applications would be better deployed using a less demanding (in terms of network and entities resources) transport communications protocol (a connectionless service, for example) either due to its functionality nature or due to network or device resource starvation. In that respect, the OSI/NMF is not flexible since it only defines a demanding connection-oriented transport layer.

The OSI/NMF entities (manager and agent) exchange management information by establishing a management communications service between CMISEs. Each peer to peer controlled management association is identified by a pair of service access points provided by the CMISEs.

In terms of management functions, the framework defines five areas of management:

- Configuration Management – provision of configuration information into object instances attributes (or properties); events may also be associated with the dynamic creation or deletion of instance objects;
- Fault Management – this area includes management procedures for monitorization of network and device faults either by polling or by unsolicited notifications; also, the framework defines alarms for monitorization;
- Performance Management – this area includes functions for specific monitorization of performance properties, like quality of services parameters; again, events and notifications can be associated with performance parameters, just like configuration of alarms;
- Accounting Management – this includes management procedures for specific monitorization and configuration mainly for deployment of accounting of network services, including traffic accounting and consumer billing;
- Security Management – this area covers management of security aspects of network services, including definition of user access policies and security measures to be deployed on network protocols of the OSI stack; this includes also the security aspects of the management services itself.

There is another less specific topic of management function for all the other management procedures (although the OSI/NMF has no real support for management procedures but for management objects and properties) that do not fall into any of the previous categories or fall into more than one of them.

2.3 Information Model

As stated before, the OSI/NMF is an information model oriented framework. It uses a pseudo object-oriented paradigm, although with some imposed conceptualization limitations and some extended features (to permit inclusion of events and notification definitions), when compared with the modern object-oriented paradigm used on many programming languages. The network services, network and device resources to be managed must be abstracted as management objects. The management agents must implement instances of these concepts. Objects have attributes that can be monitored (or accessed) or set by the managers. They can also have actions (or methods) associated that can be executed remotely by the managers, or associated events or notifications definitions. Furthermore, the objects can be structured in classes and inheritance relationships can be assigned.

Another important feature is that there is possible to explicitly establish relationships between management data, that is, between management objects. The framework even defines a set of specific and concrete types of relationships, like *is-contained-in* and *is-peer-of*.

The syntax and semantics of management objects are defined by the SMI and further explained in the Guidelines for the Definition of Management Objects. The specification of the management objects is done on MIB modules and the language used is the Abstract Syntax Notation 1 (ASN.1).

Management objects are identified by using a hierarchic identification scheme similar to the one defined by the X.500 ISO standard. The tree of identifications associated with a MIB, or part of a MIB, is known as a Management Information Tree (MIT). The MIT structure provides means for scope and filtering techniques that can be used to select specific instances of objects implemented by agents. It permits also the inclusion of authority entities together with the inclusion of real management objects. So, each object can be globally referred.

Finally, comparing the information model of the OSI/NMF with the generic conceptualization presented of the introduction of this chapter, this framework is based on a two layer information model, integrating the two lower levels of that generic conceptualization, that is, it provides a management data model where objects are defined in MIBs and a representation, encoding and transmission model, where the Basic Encoding Rules (BER) are used.

2.4 Additional Considerations

The deployment of the OSI/NMF was much constrained by the success of the OSI framework itself, since this management framework depends on the implementation of the entire OSI framework for its deployment. From this, it is obvious that the OSI/NMF deployment suffered with the decreasing support for OSI products and the exponential grow of support for Internet solutions.

Despite of this, the framework has great conceptual and functional merits, even if its dependency on a specific information model with only middle and lower layers limits its usage for modern network services distributed management. It is still a much scrutinized management framework and keeps inspiring some architectural or functional aspects of other recent frameworks, like the NSMF as a matter of fact. Also, there was additional research work on the OSI/NMF aiming to resolve some of its main limitations, like definition of support mechanisms for better deployment of distributed management techniques. It is the case of some new management functions: the Management Knowledge (X.750), the Management Domain (which includes specification of administrative policies), Command Sequencer (function for scheduling of management procedures at the agent) and the Enhanced Event Control function for added control features for events and notifications.

Further reading on the OSI/NMF, detailing the aspects introduced here, can be found at [3,14,22,29,49].

3 Internet Network Management Framework

About at the same time the OSI/NMF was developed, the Internet counterpart created the Simple Network Management Protocol, directly based on the Simple Gateway Management Protocol (SGMP),[i] but inspired in many regards on the OSI/NMF, like the agent/manager paradigm and the information model.

In the late 80's, there were three major trends for Internet network management:

- The SGMP project, which was developed with great concerns for implementation costs and quick and effective deployment on real systems, so they made it very simple;
- The CMIP Over TCP/IP (CMOT) project, which had the ambitious objective of developing a similar management framework to the OSI's solution, but over the Internet transport and network protocols; and
- The High-Level Entity Management System (HEMS) project,[ii] which was, like the OSI/NMF, a complex management framework but with a much less public support than the OSI/NMF.

The Internet Activities Board (IAB) decided in the beginning of 1988 that members from the three projects would form a task force for definition of the new Internet management protocol, based on one of the three existent solutions. Again, much emphasis was made on a short development schedule so

[i] RFC 1028.

[ii] RFC 1024.

effective implementations could be tested and incorporated into commercial products in a short period (two to three years, maximum).[i] So, marketing was already a great force behind the creation of the SNMP…

As a result of this task force, the SNMP became the recommended protocol for network management on the Internet in 1989, with a fast growing number of network devices manufactures complying with this new technology… Not surprisingly, though…

Another recommendation of the referred working group was that the SNMP based approach would be provisory and that a long term solution based on CMOT should be pursued. Furthermore, the HEMS project would have to be abandoned. Well, the true is that the CMOT project has been abandoned also and that the provisory solution has became the long term solution. This is due to many factors but the author lists the ones he considers the most important:

- The SNMP wide support and deployment from all major network devices manufactures, which makes a new solution less desirable; that is, the marketing factor with SNMP is still very high so why bother?...
- A new solution based on CMOT would be incompatible with the majority of the installed network management platforms;
- A new solution would probably be more complex for the agent side, which would raise significantly the development, testing and effective deployment costs for the majority of the companies represented on IAB working groups or task forces;
- A great research effort has been developed in the last decade (or more) for enhancing the SNMP architecture and functionalities, even from non-commercial institutions; a drastic turn to a new solution would have made those efforts less relevant; and
- Until now, no other complete framework seemed to have a good enough compromise on all important aspects that could appeal to both the scientific non-commercial community and to the commercial enterprises community.[ii]

Since its creation in 1989, the SNMP and related documents evolved and the first set of RFCs to be considered the Network Management Framework for TCP/IP-based Internets was publicly available in 1990. Some important documents were added in 1991 and 1992 to form the first version of the Internet-standard Network Management Framework (or just Internet Network Management Framework).[iii] It should be noted also, that, in contrary from what is commonly assumed, the present security model of the INMF was introduced and defined on its first version through RFC 1352, although its effective deployment was actually never obtained until version 3.

The second version of the SNMP became a proposed standard on April 1993. This new version was an overall set of documents revising the first version of the INMF and some other documents with new features were added to the framework. This brought an overall improvement on the structure of the documentation and the new version of the SNMP, together with the features and some other architecture evolutions made the entire framework, now designated as the Version 2 of the INMF (or INMFv2) more coherent and formal. This version was lastly revised and complemented on January of 1996 by a new set of documents (the second version documentation spans across a three year period, from 1993 up to 1996). Note the experimental RFCs 1909 and 1910 that defined already the MIB view-based access model and user-based security model to be applied to the INMFv2 (known as the SNMPv2u, while the standard SNMP community-based was known as the SNMPv2c)! These security approaches gained endorsement of the IETF only on the third version of the INMF.

Finally, the third version of the INMF appeared as an even bigger set of documents on 1999,[iv] being its major achievement the inclusion of a more complete and formal security model and an access model. The 1996 documents (used for both version two and version three of the INMF) were again revised on

[i] This was an obvious consequence of having a majority of representatives of commercial enterprises on this working group.

[ii] And in honesty, the author does not claim that, in the near future, the NSMF will be capable of that…

[iii] The term Internet-standard Network Management Framework appeared first on RFC 1270, from October 1991. On some IAB/IETF documents is also termed SNMP Version 1 Management Framework (SNMPv1).

[iv] In fact, other intermediate RFCs were produced in January 1998 (RFC 2260 to RFC 2266) but their standard versions were presented only 1999.

2002 and some new ones were also added. So, at present the documentation for Version 3 of the INMF has valid documents that span across a three year period, from 1999 up to 2002.

Table 2 lists all the major documents that define each version of the INMF.[i] In addition to these documents, an enormous list of related RFCs has been produced, with many becoming proposed standards. Almost all of these are definitions of management objects for specific types of network devices, protocols or applications.

Document	Title	Date
	Internet Network Management Framework Version One	
RFC 1155	Structure and Identification of Management Information for TCP/IP-based Internets (SMI).	May 1990.
RFC 1156	Management Information Base for Network Management of TCP/IP-based internets (MIB-I).	May 1990.
RFC 1157	Simple Network Management Protocol (SNMP).	May 1990.
RFC 1213	Management Information Base for Network Management of TCP/IP-based internets (MIB-II).	March 1991.
RFC 1215	A Convention for Defining Traps for use with the SNMP.	March 1991.
RFC 1270	SNMP Communications Services.	October 1991.
RFC 1351	SNMP Administrative Model.	July 1992.
RFC 1352	SNMP Security Protocols.	July 1992.
RFC 1353	Definitions of Managed Objects for Administration of SNMP Parties.	July 1992.
	Internet Network Management Framework Version Two	
RFC 1441	Introduction to the second version of the INMF.	April 1993.
RFC 1442	SMI for SNMPv2.	April 1993.
RFC 1443	Textual Conventions for SNMPv2.	April 1993.
RFC 1444	Conformance Statements for SNMPv2.	April 1993.
RFC 1445	Administrative Model for SNMPv2.	April 1993.
RFC 1446	Security Protocols for SNMPv2.	April 1993.
RFC 1447	Party MIB for SNMPv2.	April 1993.
RFC 1448	Protocol Operations for SNMPv2.	April 1993.
RFC 1449	Transport Mappings for SNMPv2.	April 1993.
RFC 1450	MIB for SNMPv2.	April 1993.
RFC 1451	Manager-to-Manager MIB.	April 1993.
RFC 1452	Coexistence between version 1 and version 2 of the INMF.	April 1993.
	Internet Network Management Framework Version Two – 1996 Revision	
RFC 1901	Introduction to Community-based SNMPv2.	January 1996.
RFC 1902	SMI for SNMPv2.	January 1996.
RFC 1903	Textual Conventions for SNMPv2.	January 1996.
RFC 1904	Conformance Statements for SNMPv2.	January 1996.
RFC 1905	Protocol Operations for SNMPv2.	January 1996.
RFC 1906	Transport Mappings for SNMPv2.	January 1996.
RFC 1907	MIB for SNMPv2.	January 1996.
RFC 1908	Coexistence between version 1 and version 2 of the INMF.	January 1996.
RFC 1909	An Administrative Infrastructure for SNMPv2.	February 1996.
RFC 1910	User-based Security Model for SNMPv2.	February 1996.

Table continues on next page…

Table 2: Internet Network Management Framework documents.

[i] Note that some documents are not standards but only informational or experimental documents, but that the author of this thesis feels they were important for the definition and understating of the various versions and evolutions of the INMF.

Table continuation…

Internet Network Management Framework Version Three		
RFC 2570	Introduction to Version 3 of the INMF.	April 1999.
RFC 2571	An Architecture for Describing SNMP Management Frameworks.	April 1999.
RFC 2572	Message Processing and Dispatching for SNMP.	April 1999.
RFC 2573	SNMP Applications.	April 1999.
RFC 2574	The User-Based Security Model for SNMP.	April 1999.
RFC 2575	View-based Access Control Model for SNMP.	April 1999.
RFC 2578	Structure of Management Information Version 2 (SMIv2).	April 1999.
RFC 2579	Textual Conventions for SMIv2.	April 1999.
RFC 2580	Conformance Statements for SMIv2.	April 1999.
Internet Network Management Framework Version Three – 2002 Revision		
RFC 3410	Introduction and Applicability Statements for INMF.	December 2002.
RFC 3411	An Architecture for Describing SNMP Management Frameworks.	December 2002.
RFC 3412	Message Processing and Dispatching for SNMP.	December 2002.
RFC 3413	SNMP Applications.	December 2002.
RFC 3414	The User-based Security Model (USM) for Version 3 of the SNMP.	December 2002.
RFC 3415	The View-based Access Control Model (VACM) for Version 3 of the SNMP.	December 2002.
RFC 3416	Version 2 of the Protocol Operations for the SNMP.	December 2002.
RFC 3417	Transport Mappings for the SNMP.	December 2002.
RFC 3418	MIB for SNMP.	December 2002.
RFC 3584	Coexistence between Version 1, Version 2, and Version 3 of the INMF.	August 2003.

Table 2 (continuation): Internet Network Management Framework documents.

There is a group of IETF documents (standards, experimental or informational RFCs) that define extended mechanisms for the INMF, being the most important the documents dedicated to deployment of distributed management and policy management techniques to the standard INMF. These documents will be referred later on next sections when this framework is further analyzed.

3.1 Architecture

The architecture of the INMF is simple and is based on the agent/manager paradigm of the OSI/NMF. It is commonly known as the Administrative Model and has suffered some evolutions from since the first version of the INMF. Initially, agents and managers were associated in Management Communities with each community being identified and accessed with a community string. Each management community was an association of one agent and several managers and the community string could be though as a secret password between one agent and several managers, although it was transferred on SNMP protocol data units without any confidentiality assurances over the network.

The agent was a management entity that would have to implement instances of management objects that represent network and device resources abstractions. Proxy agents (both native and foreign) were concepts already defined. The manager, also named Network Management Station (NMS) had to implement the management applications that would interact with the managed nodes using SNMP. Management activities were realized by means of monitoring and setting of values of the management objects instances and by using notifications on pre-defined events, called SNMP traps. The functionality of these traps was very limited and the most used method for deployment of management procedures was through polling techniques,[i] which, in conjunction with the over-centralized, non-hierarchic architecture narrowed the framework utility to effective low level monitorization management.

[i] It was created a special MIB for remote monitorization of local area network protocols and interface statistics: the Remote Monitoring MIB. Its first version was defined on RFC 1271 (November 1991), later revised on RFC 1757 (February 1995), and its second and last known version was defined on RFC 2021 (January 1997), with important extensions on RFC 3144 (August 2001).

The 1992 extensions to the first version of the framework, with the formal definition of the SNMP Administrative Model on RFC 1351, introduced the concept of SNMP Party as the entity (or process) responsible for implementation of a SNMP agent or SNMP manager on a network device, ending the appliance of the community conceptualization on the framework. Each network device could execute several processes implementing several SNMP parties, each with a set of associated attributes that would define its access policies, network addresses, transport mappings and security parameters. In the opinion of the author of this thesis, the conceptual evolution preconised on these three 1992 documents were the most important single evolution step on the entire lifespan of the INMF until now.

The second version of the framework didn't bring any real semantic changes on the revised INMFv1 architecture. The most important architectural evolutions were the introduction of SNMP Context concept and the added extra functionality level provided by the inclusion of the possibility for two managers to communicate to each other, although limited to a simple form of information requests (that expected an associated report form the target manager). A SNMP context was the equivalent to the former community concept of the INMF, but applied to SNMP parties. There was also the introduction of SNMP Messages Classes that, together with the SNMP party and SNMP context concepts permitted the construction of SNMP access policy tables.

Despite this evolution, the INMFv2 would also permit the implementation of community-based agents to be INMFv2 compliant (which would become known as SNMPv2c). Many, like this author, believed this was a step backwards, only included to permit that many network device manufactures could claim that their products were SNMPv2 compliant only by adding smaller changes to the SNMP protocol itself without architectural changes, implementation of access policies or security mechanisms.

For a more detailed analysis on the architectural differences between the first and second versions of the INMF please check RFC 1908 and [22,30,123,127].

The third version of the INMF incorporates no real innovations; the SNMP party concept was abandoned and substituted by the generic SNMP Entity concept (although SNMP contexts remain to be used). A SNMP Entity contains a SNMP Engine implementation and some SNMP Applications components. The engine part is responsible for the entity's identification[i] and for all components common to all SNMP entities, agents or managers, including the SNMP message format;[ii] these components were introduced on RFC 2572 and revised on [125]. The SNMP applications part is formed by several sub-components[iii] that, depending on its implementation or not, defines the entity as a SNMP agent or a SNMP manager; this component was defined on RFC 2573 and later revised on [124].

Security & Access Control Models

On the security aspects of the framework, it can be said that the first version of the INMF already presented important work on this matter through RFC 1352. At this time, only a security model was presented with no real formalization of an access model. The security model integrated three end-to-end security services (data integrity, data origin authentication and data confidentiality services) for assurance of confidentiality, authentication of the entity originating the SNMP message and detection of modification of the SNMP message content. Additionally, this RFC defined a complex mechanism that could provide some control (or notion) of the time of creation of each SNMP message; this was named a message timeliness mechanism. Nevertheless, this security model, applied to SNMP parties never got to be effectively applied and endorsed by SNMP implementations of the time.

[i] That is, the engine identification is also the entity identification in the sense that there is a one-to-one relationship between a SNMP entity and its SNMP engine.

[ii] These include the SNMP message processing and dispatcher, the security and the access control sub-systems.

[iii] The components of the applications part of a SNMP entity can contain a command generator and responder, a notification receiver and originator, a proxy forwarder and, eventually, other non standard applications. Clearly, a SNMP agent must implement a command responder, a notification originator and proxy forwarder, while a SNMP manager must implement a command generator, a notification receiver and a notification generator.

Later on, version two of the INMF lightly revised the previous document through RFC 1446. On the other hand, RFC 1910 presented an important revision of RFC 1446, since it is a much more complete document (presenting numerous implementation considerations), although there is no major addition of new security mechanisms or security functionalities. Nevertheless, it should be noted the introduction of the basic concept of time window (the period of time each SNMP message was to be considered valid for processing) and the term User-based Security Model (USM) for naming the model. As stated before, deployment of this security model was never obtained (the framework with this security model was known as SNMPv2u) and the endorsed version two of the INMF even contained documents that permitted and encourage the use of an unsecured SNMP (aka SNMPv2c).[i]

Finally, the third version of the framework presented another important revision of the existent security model through RFC 2574, although this was, again, mainly a major document re-writing, describing in greater detail the functionality and specification of the USM and the integrated security mechanisms. More importantly, a new View-based Access Control Model (VACM) was presented (RFC 2575), re-utilizing the older MIB view concept, though. The last revision of these two models was introduced by [131] and [132], respectively, but these were not extensive revisions and presented only some format corrections and terminology normalization.

More detailed considerations on the security and access control model of the INMF will be made when presenting the EACM on Chapter Four, or when evaluating both frameworks on Chapter Seven.

An extensive description of the overall architecture of the present version of the INMF is presented on [123]. This author feels that many parts of the standard documentation defining this last version of the INMF architecture are detailed in excess, that is, the exaggerated concern on presenting such a modular perspective of the internals of SNMP entities is of no great use for the overall community and presents the framework from an implementors view. In this author's opinion, this type of concern should had been put when describing the interaction between entities, that is, when specifying all management protocol or services interfaces, and not when describing the internal architecture of management entities.

3.2 Communications & Functional Models

The INMF communications model has evolved little from version one up to the present. It is based on a single type of management communications service provided by SNMP. In fact, not few can recognize the framework only as the SNMP Management Framework (even the standard documents use both names for identification on the framework), although SNMP is only one of its components.

This protocol is intentionally simple; it uses only one basic Protocol Data Unit (PDU), and provides an even smaller set of primitives and with more limited functionality, than the OSI/NMF. The SNMP primitives can be divided into three types of primitives, based on their security aspects and management functionalities:[ii]

- Passive – this kind of primitives is used to gather, or poll, management information from SNMP agents, and thus, effectively implement monitorization management;
- Active – this kind of primitives serves to actively change the values of the management object instances on the agents, and thus, effectively implementing configuration management; and
- Unsolicited – this type of primitive is used by the agents to convey unsolicited monitorization information to managers.

The first version of the INMF integrated the first standard version of the SNMP, or SNMPv1, specified on RFC 1157. This version defined three passive primitives (SNMP-GET, SNMP-NEXT and SNMP-RESPONSE), one active primitive (SNMP-SET) and one unsolicited primitive (SNMP-TRAP). The SNMP-RESPONSE primitive is used by the agent to respond to the SNMP-GET, SNMP-NEXT and SNMP-SET requests from the manager. The use of the manager's passive primitives limited the

[i] There was a third flavour of INMFv2, known as SNMPv2*, which was like a SNMPv2u enhanced version with some additional proprietary security mechanisms but got even less support, as expected, and never reached a RFC status.

[ii] The INMF has class definitions based on PDU fields (read, write, response, notification and internal classes) or on communications aspects (confirmed and unconfirmed class), but the classification defined here seems more appropriate to the present analysis.

management monitorization to small sets of values of objects instances since each value would have to be polled individually. The unsolicited primitive had a low functionality because the trap could not contain an appropriate level of information to the manager. Finally, the active primitive had also little deployment due to a lack of implementation of effective security mechanisms.

The second version of the SNMP protocol (or SNMPv2), integrated on INMFv2, was defined on RFC 1448 and added one passive primitive (the SNMP-BULK, equivalent to one SNMP-GET followed by a pre-defined number of SNMP-NEXT) and one unsolicited primitive (SNMP-INFORM) for exclusive use between managers.[i] Also, the initial SNMP-TRAP primitive was retouched to a new version SNMPv2-TRAP.[ii] The revised SNMPv2 on RFC 1905 had no functional or conceptual evolutions and it was a mere re-writing of RFC 1448.[iii]

The third version of the INMF inherited SNMPv2 as its management communications protocol that would suffer another minor revision on [130];[iv] again, no functional or conceptual revision occurred, only document correction and terminology normalization with the rest of the documents of the third version of the INMF. The INMFv3 introduced two complementary informational documents to the SNMPv2 definition: RFC 2570 and RFC 2571 later revised into RFC 3410 and RFC 3411.

The fact that the third version of the INMF still uses the second version of the SNMP is another reason justifying this author's preference for usage of the INMF naming instead of the SNMP Framework term, although the later is the most used terminology on IAB/IETF documents or even by the majority of the scientific community presenting research projects on this area.

The INMF defines no mandatory transport, network or link protocol for encapsulation of the management communications protocol, although it recommends the use of a connectionless transport protocol, namely the Internet User Datagram Protocol (UDP). The usage of other types of transport protocols have been more seriously addressed on RFC 1449 and guidelines for encapsulation on other connectionless transport technologies were added: SNMP over the OSI's connectionless transport service, SNMP over AppleTalk's Datagram Delivery Protocol and SNMP over Novel's Internetwork Packet Exchange protocol.[v] This RFC was lightly revised on RFC 1906 and on [129].

Lately, there has been a greater concern to provide guidelines for usage of SNMP over connection-oriented transport protocols, namely the Internet Transmission Control Protocol (TCP). The major advantages for use of this type of transport protocols are the support for a more efficient transmission of large streams of management data and a much greater reliability due to its support for management data flow control, including indirect confirmation of reception of SNMP PDUs. The major difficulty to use a connection-oriented transport protocol comes from the fact that the SNMP, as a management communications service, is inherently connectionless. Nevertheless, important work has been already presented on [128].

3.3 Information Model

Like the OSI/NMF, the INMF is also an information model-based management framework; it defines the two lower levels of the reference model presented earlier on this chapter's introduction: the middle layer is occupied by a management data model defined by the MIB concept, firstly introduced on the INMF by RFC 1156 and immediately revised by RFC 1213, and termed MIB-II. This document defined the

[i] The SNMP-INFORM primitive is equivalent to a SNMP-TRAP primitive but is for use between managers (or SNMP applications).

[ii] No real functionality was added, but a more detailed and formal description of this type of primitive is made. Also, this new version of trap would use the same basic SNMPv2 PDU that all the other primitives use, whereas the first version was coded on a special SNMP-TRAP PDU.

[iii] At this time a new primitive (SNMP-REPORT) was introduced but no standard meaning was defined to it.

[iv] Only the most important RFCs of the present version of the INMF, together with other recent and relevant RFCs, that extend its functionally or attenuates some of its major limitations, are included as reading references on this thesis. The other RFCs explicitly listed on this section are presented for completeness and for historic reasons.

[v] A companion document (RFC 3419) with textual conventions was created specifically for aiding in the definition of different transport protocols addresses using standard SMI syntaxes.

standard MIB with a set of common management objects for management of the TCP/IP protocol stack and other common link layer interfaces monitorization. Since then, a large number of specific MIBs for management of specific Internet services and applications, proprietary network devices and all kinds of interfaces technologies were created.

Probably the most important MIBs are:

- The MIB for management of SNMP entities, that is, for management of SNMP services itself; this MIB was introduced on RFC 1450 of INMFv2 and revised on RFC 1907; its last version was presented on [126] for the third version of the INMF; and
- The Remote Monitoring MIB (RMON)[i] for gathering of monitorization statistics from remote probes or monitors on the local area network; the RMON, with a first version on November 1991 (RFC 1271) and a second version defined on 1997 [36]; complementing or extension MIBs were defined on RFCs 1513,[ii] 2074[iii] and 2613,[iv] and, more importantly, on [78,102].

The entire functionality of the framework is limited by the functionality boundaries set by the MIB object concept, which is much more restricted than the MIB object of the OSI/NMF. On the INMF, management objects do not have attributes and it is not possible to directly associate methods (or functions) or events to objects. Furthermore, it is not possible to define relationships between objects or object attributes. Since the management data model of the INMF is based on objects with restricted functionality levels, the framework can not be used for management of higher functionality network services. Its limited power of abstraction is only adequate for low level functionality management of local network device resources.

The MIB objects are identified using Object Identifications (OID), using the same hierarchic identification space of the OSI/NMF. Although this is a good design decision, the INMF does not have the scope and filtering capabilities of management objects. If we add the lack of standard compression techniques for OIDs, makes the use of OIDs less attractive in the INMF than it was on the OSI/NMF.

The INMF management data objects are defined on MIBs using a language defined on the SMI (a sub-set of the ASN.1 language), firstly defined on RFC 1155 and lightly revised on RFCs 1442 and 1902. A second version of the SMI was defined on RFC 2578, which added some object types and simplified some others, although with no improvement on the overall functionality. This version was not revised since. Complementary documents with standard textual conventions were introduced on INMFv2 with RFCs 1443 and 1444 and later revised on RFCs 1903 and 1904. Their last revision was made on the INMFv3 through RFCs 2579 and 2580, respectively. These textual conventions and conformance statements provided an added functionality to the low level types of objects permitted by the standard SMI.

3.4 Recent Enhancements

Although it has made the greater impact on marketing terms, the security enhancements of the USM and VACM definitions of the third version of the INMF are not new, not even recent. It has been pointed out previously that the concepts and mechanisms behind these models were defined already on the first version of the INMF (although we different naming) and revised for INMFv2.[v] If it were not for the conservative strategy adopted by the IETF and these models could have been deployed sooner. What is very revealing is that almost all security or access mechanisms integrated on the first proposal for the INMFv1 were maintained as the recommended mechanisms up to the last revision of the USM and VACM on 2002.[vi] In this respect, an important ongoing project must be pointed out: the Session-Based Security Model (SBSM), defined on [163], justified by the need for a session oriented management

[i] This MIB should not be considered as a central document for definition of the INMF but represents an important architectural evolution in the sense that it represents a sort of delegation of some processing power, and thus, execution of some management procedures to the agent implementing the RMON MIB.

[ii] RMON Token Ring Extensions.

[iii] RMON Protocol Identifier.

[iv] RMON for Switched Networks.

[v] An alternative approach for security deployment on INMF was proposed on [39]. This proposal was based on existent secure transport technologies, like the Secure Sockets Layer service, that would imply no direct change to the INMF architecture and SNMP protocol.

[vi] By now, some of the cryptographic methods could have been substituted by alternative, more recent algorithms.

communications service and the introduction of session keys, in opposition to the present oversimplified INMF key concept; some of these concerns coincide with some of the requisites for the NSMF communications model.

This is to say that other recent developments seem to have a comparable potential for a real impact on the effective deployment of the INMF, when compared to the USM and VACM. This is the case of research work on two relevant fields for modern network services management: distributed management and policy management.

Distributed Management

Overly centralized network management frameworks are prone to scalability problems on this age of distributed network services and applications. This, and the lack of high functionality levels of management procedures, makes the INMF an inappropriate recipe for an effective deployment of correct distributed management technologies and mechanisms.

Nevertheless, some relevant work has been done on the last decade to help the INMF to attenuate its limitations for distribution of management functionality over multiple management entities (unfortunately, present INMF architecture can not support the concept of distributing management functionalities over multiple management services). This started with two relatively old MIBs: the already referred RMON MIB and the Manager-to-Manager[i] (M2M) MIB; but, while the later is a good instrument for low level distributed monitorization of local area networks, the former never really got enough endorsement due to its overly simplistic nature and was later substituted by other efforts on the same area from the responsibility of the IETF Distributed Management Working Group (DISMAN).[ii]

The most important work of the DISMAN was the creation of MIBs for implementation of some distributed management techniques:

- Alarms – this feature permits the remote configuration of management agents to implement alarms, that is, a local continuous monitorization process that can compare a set of object instances values with a group of threshold values and set the associate alarms when these threshold values are attained. The important evolution here is that the process of monitorization and alarm setting is delegated from the SNMP manager to the SNMP agent. This concept was firstly introduced on the INMF by the RMON MIB and, later on, gained enough importance to deserve specific documentation. The last version of the documents defining SNMP Alarms is presented on [160,161].
- Events – these and alarms are complementing and overlapping features that form an essential part of any distributed management technology since it permits the management servers (SNMP agents in the case of the INMF) to inform, in an unsolicited way, the management clients (SNMP managers in the INMF) of pre-define events[iii] that the servers could detect. The DISMAN definition of this feature for the INMF is made on [90]. The true is that INMF events and alarms convey, more or less, the same concept and this author feels that both approaches could be united on a common feature, being this the event, because this concept seems to be more generic. This is the justification for the definition only of events on the NSMF, since its definition includes all features and functionalities provided by alarms. On the INMF, though, the research work on alarms is, at this time, more advanced than on events, so this could become the chosen long term feature. Alarms and events generate notifications (or traps)[iv] that can be logged on the Notifications Log MIB [91].

[i] RFC 1451, April 1993.

[ii] The SNMP Research International, Inc. enterprise has an undergoing project (as Internet Drafts to be discussed on the DISMAN) for definition of a SNMP Middle Manager MIB and associated SNMP Script Language that could raise relevantly the functionality of this concept of a SNMP dual role management entity.

[iii] Optimally, an event can be triggered by any pre-defined combination of a set of conditions of a group of monitored management objects (or services and applications).

[iv] Notifications syntax and semantics are specified on the Notification MIB table, which is defined on [124].

- Remote calculation of expressions – this feature, defined on the Expression MIB [89], permits the computation of boolean expressions for monitorization of alarm thresholds and event triggering conditions;
- Delegation of Script MIBs – this advanced technique defined on [101] can be used to delegate management procedures from a SNMP manager to another SNMP entity (agent or manager, but the term distributed manager[i] will be used to distinguish it from the manager delegating the script code) that implements the Script MIB. Nevertheless, this DISMAN solution has some important limitations: it does not define or recommends any standard language or encoding techniques for the script code; it does not define a security model to be applied when dealing with remote execution of management scripts and access control must be implemented using the VACM approach;[ii] the identification scheme of the scripts is implementation dependent; the parameterization of the script execution must be made offline;[iii] the system provides a push method for transfer of the script code from the SNMP manager to the distributed manager using standard SNMP table manipulation techniques (which are complex, by the way) and a poll method from the distributed manager which involves an Hypertext Transfer Protocol (HTTP) or File Transfer Protocol (FTP) access after the SNMP manager has informed the distributed manager of the Uniform Resource Locator (URL) of the script code (which seems an odd and unnecessary alternative for transfer of the script code, with additional security and device resources consumption implications); it is not possible to re-use scripts already delegated by other scripts; finally, the entire process must be managed remotely by the SNMP manager using complex MIB tables manipulations, which tends to discourage network management applications implementors.
- 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 abstract the scheduling of management scripts if the management entity implementing the SNMP scheduler also implements the script MIB. The functionality of this feature is defined on a specific MIB [57] and is only adequate for scheduling of simple management tasks or actions that are easily modelled as integer values. Scheduling of more complex management procedures or combination of time and conditional scheduling is not practical due to its implementation complexity or conceptually unsupported.

These features definition is the most relevant and visible work of DISMAN. Further analysis about the development and usage of these MIBs can be found on [71] and [76]. This last reference also discusses the mobility, which is another important aspect of distributed management. This specific are of distributed management has seen much less research efforts when associated to the INMF. One exception is the JAMES project described on [69], which describes a SNMP, JAVA and CORBA integrated approach on management agent's mobility.

A last word on a special DISMAN MIB [87] devoted to provide means for remote execution of some traditional TCP/IP network management utilities: Ping, Traceroute and Domain Name System lookup. This MIB is referred here because it clearly shows the low functionality level provided by the information model supporting the INMF. Even the simple task of executing remotely such common Internet management utilities using SNMP, involves the implementation and manipulation of an extensive and complex MIB. This MIB seems like an exercise on the art of creating and writing MIB modules with no relevant usefulness…

A good resource for discussion on how to enhance the INMF capabilities for distributed monitorization using a hierarchical structure of SNMP entities is provided by [27]. A similar approach but proposing a more complete hierarchical organization of SNMP entities (including Manager MIBs) is described on [168].

[i] This term is not particularly appropriate but it is the DISMAN official term so it will also be used here.

[ii] This obligates the script's structure on the script MIB to be adapted to this low level access control model.

[iii] The instance values of the script MIB objects representing the script arguments must be set first and then the script is triggered for execution. This must be done every time the script must be executed.

Policy Management

Apart the work of the DISMAN, some important research has focused on policy management using SNMP. Since the information model of the INMF only defines a middle layer (that is, a management data model) with limited abstraction power, it is difficult to support policy management directly with SNMP and MIBs.

Policy management support on the INMF has been addressed by some research projects outside IETF and by the IETF Configuration Management with SNMP Working Group (SNMPConf). [63] describes the overall strategy followed by SNMPConf to address this issue and its work has produced two configuration MIBs (RFCs 3512 and 3747) but that are not directly related to policy management. In that respect, an Internet draft implementing a policy management MIB is the most relevant SNMPConf ongoing project.

Another important effort is made by the Network Management Research Group (NMRG) of the Internet Research Task Force (IRTF) [169]. The NMRG has been working on a new language for definition of a higher layer of the INMF information model. This language, known as SMI for the Next Generation (SMIng) should permit the specification of policies to be translated into SNMP MIBs (specified using SMIv2) and also into other management data models like, for example, Policy Information Bases (PIB) specified using a sub-set of SMIv2 named Structure of Policy Provisioning Information (SPPI). This new language is defined on [156] and [111] is an interesting and complementing reference. Finally, [155] provides the methodology for mapping SMIng into SMIv2.

Outside IETF or IRTF working groups some interesting work has been developed focusing on implementation of technologies and mechanisms that can enable the INMF to better suit policy management. This author would like to emphasize reference [59] that describes a methodology, and its limitations, for deployment of policies using the present INMF version without need for an extended information model, but at a cost of extended complexity and reduced performance. Another interesting work on an upper layer information model named Simple Network Management Knowledge is described on [45].

3.5 Main INMF Limitations

The technological efforts developed for the INMF evolution try to use indirect mechanisms to provide some of the mid or high-level functionalities lacking on the original framework. The problem with the majority of these mechanisms is the integration with each other and with the low-level functionalities already available, leading to an excessive number of indirect objects to manage and complicated procedures of manipulation. In some extent, this is also truth to the model and mechanisms used to implement and provide security and access control.

Some of the listed references [4,15,22,30,71,118,164] directly identified some main limitations of the model and mechanisms of the INMF, but many of the other listed resources on this thesis list of references pointed out other more specific vulnerabilities. This author has already contextualized some of those constraints in the previous sections of this chapter, and the following list summarizes what the author feels are the most relevant limitations of the INMF that motivated the development of the new NSMF:

- Scalability problems due to the over-centralized manager/agent and the data polling paradigms of its architecture. This almost flat client/server architecture is very limited for creation of various levels of management since the objects represent local resources on an individual network device. The problem has been identified since the INMF creation but no effective and viable alternatives have been deployed.
- The framework is constrained to a rigid, low level functionality Information Model that only integrates a middle-level management data model and a low-level encoding and transmission model.
- Lack of high-level functionalities available for building Network Management Applications. This is due to the excessive simplicity of the semantics associated with MIB objects.
- Low efficiency of the model regarding the large amount of non compressed raw data transferred between managers and agents, since the processing of the raw data has to be done mainly on the managers/management application side. Also, object manipulation, like tables,

is somewhat complex because the managers can only rely on very basic, low-level procedures/methods. So, to accomplish something more complex than trivial manipulation, the manager has to issue a group of related low-level operations, all of them transmitted to the agents. Generally, all these operations have individual responses that are also transmitted through the network, increasing even further the impact on network bandwidth, which tends to be the most expensive resource on a distributed system.

- Complexity of the access control and security procedures of the model standardized on the USM and VACM models. Despite being powerful, these mechanisms still use complex chains of small operations and need a large management burden for their extensive list of associated objects.
- The protocol operations conveyed on SNMP PDUs are always dependent on the underlying transport protocols of productive protocol stacks to be transferred between management entities. This limits network management availability on situations where these transport protocols are unavailable[i] and complicates network traffic and quality of service management because mandates that network management services traffic to be included on the global network traffic classification scheme.
- There is no support for naming of the entities so any changes on the management entities addresses must be propagated to the local configuration of all the other entities using an external mechanism to the INMF, probably manual configuration.
- Except for some objects on MIBs dedicated to network monitoring, the managers must process the majority of the raw management data and make all the micro[ii] and macro[iii] management policy decisions, including the adaptation of configuration changes made through the limited and complex manipulation of MIB objects. Again, this brings relevant scalability problems and high development and implementation costs to network services management applications.
- The key management system of the USM is limited to a remote method for updating an existent user key. The key concept is very incomplete since it does not incorporate any ownership, security model relationships and timing parameters. The method for key localization can be a security liability since a unique user key gives access to all its associated localized keys.
- The levels of access control permitted with by the VACM are very limited and the interface for its applicability (through MIB views) is complex. The INMF framework does not have support for management entities and network resources consumption control, which could result on entity or network resources starvation, either relative or absolute.[iv] This could be considered a security liability since resource starvation could be induced on purpose by a single entity or group of entities.

As stated before, some of these problems have been addressed in the past years with the creation of mechanisms that try to increase the functionality level of the managed objects and make the model less centralized. Such efforts include distributed management and management by delegation concepts that are partially implemented in the Remote Monitoring, Event, Alarm, Expression, Script and Scheduler MIBs. Nevertheless, the functionality level of the individual objects of these MIBs is still very limited and their management, from the managers point of view, complicated and resource consuming.

[i] This could be due to operational network protocols problems or just because the devices do not implement productive protocols higher than the link or network lawyer.

[ii] Micro management (or local operational management) policy deployment includes the decision making and correspondent configuration changes of local parameters per agent. It concerns about all the aspects of the agents configuration and how this affects its behaviour on a local basis.

[iii] Macro management (or global operational management) policy deployment includes the decision making and correspondent configuration changes of global parameters per group of agents and managers or per management domain. It concerns about all the aspects of the network protocols global configuration options and how this affects the behaviour of overall network services and applications.

[iv] Relative starvation refers to the fact that a small group of entities, generally a manager, consumed, at least, one type of resource on an agent or on the network when executing a specific management procedure but the majority of the productive protocols on the agent and the network can still operate, while absolute starvation means that resources on the agent or the network were consumed to the point to compromise the operational status of the agent or/and the network.

In the approach presented on this thesis, these concepts of distributed management and management by delegation are implemented using the same integrated mechanism in the NSMF model and can be provided directly to the user as Services Management Functions. Furthermore, this new framework is intended for the management of network services and distributed applications and uses specific and internal management communications protocols (either connection-oriented or connectionless) so, if needed, the framework can be deployed on top of a simple transport protocol or even directly on top of the network layer.

3.6 Additional Considerations

The INMF, and ways of adapting it to modern network management requirements, continues to be a major subject for research work on the network management fieldd. Although less important than distributed management or policy management, other aspects of management of modern network services has been addressed and one of the most relevant (apart from the distributed and policy management) is the integration of SNMP technologies with OSI/NMF technologies. Reference [31] describes a framework for deployment of part of one framework using the other framework (CMIP over SNMP and SNMP over CMIP) and how to use CMIP and SNMP simultaneously on the same network management platform; it discusses also the major advantages and limitations of both frameworks. The same type of research, with a more detailed explanation, but considering an older version of the INMF, is presented on [12]. Finally, a recent and complementing article dedicated to integration of network management technologies endorsed by the IAB, IETF, IRTF and Distributed Management Task Force (DTMF) is presented on [147].

As a closing remark to this section, the author would like to point the research [62] on a management platform based on the third version of the INMF for management of mobile IP wireless networks, including the effective implementation of a management application. Also, reference [52] provides an invaluable resource for description of the evolution of the INMF and some discussion on the impact that some recent features added to the framework will have on the field.

4 Other Relevant Frameworks & Technologies

This section will be dedicated to introduction of other relevant network services management frameworks or technologies other than the OSI/NMF or the INMF. Firstly, some technologies of distributed management will be described (several management delegation methodologies will be addressed); then alternative management frameworks based on recent technologies will be presented (like HTTP or CORBA). Finally, research on management information and management data modelling will be referred (including policy and QoS management).

4.1 Distributed Management

On modern network services and distributed applications platforms is very important to also distribute management tasks. Management frameworks that have an overly centralized architecture (like the OSI/NMF and the INMF) are prone to scalability problems. This becomes even worse by constraining those frameworks to information or data management models with low functionality levels.[i]

On the previous section we have exposed methods and technologies for incrementing the INMF distributed management capabilities. Nevertheless, this framework is imminently limited on this respect. This evidence triggered the development of some new management frameworks and many new technologies integrating and integrated on existent frameworks. This was also the main motivation for the development of the NSMF.

Distributed management technologies can be divided into four types of well known methodologies that, if desired, can be combined on the same framework:

- Multi-level hierarchical architecture & multi-domain entity organization – this type of approach divides overall management goals into smaller lower level management tasks delegated into several management domains; this delegation process can repeat itself to lower

[i] A good work on a compared evaluation of a centralized framework against a decentralized framework for monitorization management is provided on [116].

level sub-domains on the same hierarchy, forming an hierarchical structure of management entities; each management domain is responsible for deployment of management procedures with a higher functionality level than its sub-domains; at any middle level of the management hierarchy, management entities can act in a dual role, that is, they can act in the role of a management server (also known as agents) or in the role of management clients (also known as managers).

- Intelligent management servers – this technique pushes intelligence (management capabilities and functionalities) from the managers to the agents, effectively passing some of the onus of the management activity.
- Delegation of management entities – this technology addresses the mechanisms for implementation of mobile management servers and clients, although this tends to be analysed only on the server's mobility (that is, agent mobility).
- Delegation of management procedures – this approach defines the transmission of code representing management procedures from management clients into management servers. Although this approach causes less hype than the previous technology, this author thinks that mobile management code is more feasible, cost effective and can implement de same type of functionality provided by mobile management entities. That is, the concept of mobile management entities (or mobile agents) seems impossible to implement without the need for a third party runtime environment. Furthermore, it can be mapped into management code mobility without significant losses on functionality.

This author would like to note that references [15,28,55] provide insight discussions on the conceptual differences of a centralized and a distributed approach to network management. Additionally, [28] concludes that for deployment of an effective network services management system it is necessary to leverage the use of centralized strategies with distributed techniques, and that the management frameworks should be able to provide an adequate support for both philosophies that would be deployed as needed. This coincides with this author's opinion and has influenced the hybrid architecture of the NSMF that supports a high level of management distribution but retains some level of centralization for access control mechanisms and strategic management.

Hierarchic Multi-Domain Architecture

Although an important issue, there has been not much recent work on this topic, apart from [34,96,97,114], although [114] gives more emphasis to the distribution of management by mobility of management entities. Proposal [96] integrates a multi-level hierarchic management structure with mobile agents and evaluates the level of distribution of a management framework using three classes: weakly distributed, strongly distributed and cooperatively distributed). Articles [34] and [97] present similar approaches for adding a hierarchic architecture to SNMP management frameworks.

Nevertheless, references [17,41,43] provide relevant insights on this matter, that greatly influenced the NSMF architecture. Furthermore, reference [41] presents an approach using a distributed object technology different than CORBA and [43] integrates a multi-level hierarchic management structure with mobile managers complemented with mobile agents of the newly developed management platform named MAGENTA.

Intelligent Management Servers

This technique takes advantage of the fact that in modern network platforms, devices where the management servers reside became almost as powerful as the devices or hosts where the management clients reside. This assumption facilitates the shifting of intelligence from managers to agents, distributing this way the management tasks and management decisions and making the usage of device and network resources more efficient.

It is not easier to find relevant research work for this distributed management trend than for the previous trend. The author recognizes the work on [24] to be innovative at the time it was presented and introduced the theme of intelligence delegation from managers to agents in the context of management delegation. Another important, although more recent, article on integration of intelligence delegation and management entity mobility is presented on [42] as the Intelligent Mobile Agents (IMA) architecture;

an interesting feature of this paper is that it provides a comparison study on efficiency of real management tasks implemented by the IMA and by a SNMP-based management platform.

Finally, reference [142] should be recommended due to its proposal for Autonomic management entities, that is, management entities that have enough intelligence to automatically and independently deploy configuration management as an automatic response monitorization management. This approach may be of great importance for effective deployment of policy management.

Mobile Entities and Mobile Code

Delegation of mobile entities and delegation of management procedures code are, with no doubt, the most researched topics on modern distributed management technologies. An overwhelming list of articles and other documents were written on these subjects and this author will try to present, from the numerous papers it analysed on the last decade, the ones it considers the most relevant.

It must be pointed out that the set of documents [4,5,6,9,10,11,15,24,26,48] produced by G. Goldszmidt and Y. Yemini during the 90's represents the most extensive and relevant contribution on this thematic. This set of documents address various aspects of distributed management and, more prominently, of management delegation of management entities. Their analysis addresses issues related with requisites of the information model, conceptual and formal definitions of a delegation process, implementation requirements for deployment of mobile code and mobile entities, evaluation techniques of efficiency of the distributed or delegation process, security implications of management delegation processes, integration scenarios with existent network management frameworks (like the INMF and the OSI/NMF) and integration with delegation of intelligence to management servers. Nevertheless, it should be noted that these authors endorse greater utility for mobile entities (and more importantly, mobile agents) than for mobile management code, which, as stated before, is not the opinion of the author of this thesis.

Other researchers have also developed interesting solutions for this type of distributed management and references [66,83,146,151] can be emphasized because they provide management platforms for deployment of mobile management entities over frameworks with integrated technologies that include SNMP, CMIP, JAVA and CORBA. Additionally, it can be referred that [146,148] provide integrated technologies for simultaneous support of mobile agents and policy management.

On the other hand, [21] and [44] describe management systems supporting delegation of mobile management code instead of mobile management entities. The first reference specifies solutions for management delegation using delegation of scripts on a SNMP-based framework, although at the time the DISMAN Script MIB had not been developed yet. The second reference is very important because outlines the most important reasons for considering delegation of management code more useful and feasible than delegation of management entities and addresses security and access control issues; it also points out the possibility of integration of different technologies like SNMP and JAVA to effectively implement management delegation.

References [80,143,107,152] should be recommended for an approach on management entities mobility by using a two-level mobile entity concept. These solutions, define a component-based architecture for implementing mobile agents (or even any type of management entity) where each entity is formed by two or more components at different functionality levels ([152] only defines a two level component-based mobile agent) and only one (or some) of the components is delegated. The most interesting aspect of these proposals is that they provide the architectural concepts for implementation of mobile management entities using an underlying framework that in reality only supports mobile management code.

Finally, a word on specific research work about security and access control aspects of management delegation, and in particular, associated with mobile agents. Papers [53,64,158] discuss and proposed similar solutions for implementation of security and access control features on mobile agents or frameworks using mobile management entities.

4.2 Policy Management

Support for configuration management using higher level languages representing management policies (which are abstractions of a higher level that representation of local device configuration parameters) is an essential technology requirement that must be supported by management frameworks that intend to be successful.

Management frameworks that are tied up to fixed information models should define at least one information model with capacity for a high level of abstraction. Only this can provide an effective support for deployment of policy management.

One of the most important merits of the NSMF is that it is not tied to any information model or management data model. This was intentionally left outside the scope of the definition of the framework as a way of supporting any adequate information model for policy management.

Two of the first efforts for definition of a configuration language with a higher functionality level than it was permitted by the management data model of the INMF or the OSI/NMF was presented on [13,20]. In this respect, while [20] does not refer directly to policy management, [13] defines a generic framework with direct support for policy management, which, at the time, could seem an innovative technology and awkward terminology…

At this moment it is important to introduce the work developed by the DTMF organization [167] on distributed management and, more specifically, on information models. The most relevant feature of the framework defined by the DTMF is its management data model, termed Common Information Model (CIM) and defined on [60,166], which is a much better candidate for usage as a policies specification language than the present INMF management data model based on MIBs specified with SMIv2. CIM modules are specified using a high level information model named Unified Modelling Language (UML) [171] created by the OMG.

Additionally, the IETF Policy Framework working group has defined extensions to the DTMF CIM specification for exclusive usage on policies definition. These extensions were named Policy Core Information Model and defined on [94]. Until then, the IETF most relevant work on policy management had been associated with policy provisioning by the Common Open Policy Service (COPS) Protocol. This protocol, defined on [74,95,108], is used for transfer of policy rules between Policy Enforcement Point and a remote Policy Decision Point. The policy information must be specified on PIBs that are written using SPPI [103]. It should be noted though that COPS is only a technology for transfer of policy information and does not conveys any management framework definition. Nevertheless, some approaches [84,] have proposed management frameworks for policy-based network management using COPS as the policy management technology and the INMF (or any other existent management frameworks) as the management framework supporting it. Article [157] offers a useful discussion on these technologies conceptualization and its usage.

Other proposals, outside the IETF working groups and outside DTMF, address the problem of defining an adequate information model (or programming language) for specification of policy information. Some research work follows own individual concepts, like [61,93,110,138]. The first defines the Policy Description Language (PDL), a simple but powerful and expressive language. The second specifies a declarative and object-oriented language named Ponder. On [110] it is defined the Logic-based policy specification, based on the Prolog programming language while [138] proposes a policy specification language based on the object-oriented Resources Description Framework and uses XML syntax. Perhaps the most interesting approaches have come from proposals [56,141,153] for new high level management information models based on the Ontology concept that seems extremely adequate for definition of policy information, although article [56] restricts the use of ontologies to QoS configuration management.

Many policy-based technologies are oriented for QoS network services management like the one presented on [145] that uses an advanced dynamic policy-based management technology that was experimentally used to implement QoS management. This technology used the Ponder policy language to specify adaptive policy information. A similar approach is presented on [105], but with use of management scripts for description of policies. A related proposal to these three, but using an evolutionary approach, is described on [137]. In this paper, QoS monitorization techniques are integrated with the existent IETF COPS protocol for simultaneous QoS policy provisioning and monitorization.

This section will be closed by referring some important, although more unusual, research projects. The research work described on [93,100,162] deals specifically with policy management of security services (although [93] addresses policy management on a more generic approach by defining a policy information modelling language named Ponder), that is, proposes solutions for automatic management of network services and distributed applications security policies (that can be easily extended to support access control policies). Finally, article [79] discusses the evaluation of management policies and concludes that it is a problem of an overwhelming complexity for generic policy enforcement. Nevertheless, an evaluation method is proposed that, although has some usage restrictions, it was concluded that it could yield correct results on many realistic situations. Experimental results were obtained by manipulating policies specified on PDL.

4.3 QoS Management

Although it is uncommon to find research work on specific QoS management frameworks, this author would like to recommend two reference readings that present management environments specifically defined for direct support to QoS management. The first article [88] presents two experimental resource management algorithms that could be used for QoS management. These mechanisms were implemented on a prototype management system named MetaNet and are conceptually independent of any existent management framework. On the other hand, the second article [106] proposes an integrated and hierarchic environment based on existent IETF policy provisioning and DMTF web-based management technologies for management of QoS-enabled networks, named QoS-Aware Management Environment.

4.4 Java & CORBA

These two technologies were not created for direct usage on network services management but they provide useful technologies that can be integrated into existent or new management frameworks. In general, like the NSMF, the usage of these two technologies is not mandatory or strongly recommended as mean for guarantee deployment of any relevant functionality. Instead, they are recommended as alternative technologies to pre-defined mechanisms of the framework.

Nevertheless, some interesting proposals have chosen one of these technologies for their basic ingredient when defining new or extended management systems. That’s the case of a reference proposal found on [46] for the specification of a network management system based on the Common Object Request Broker Architecture (CORBA), on top of a management communications service provided by CMIP, as a mean to inherit its object-oriented functionalities.

Article [51] proposes a management framework using the INMF management data model but using a Java-enabled browser to act in the role of management client and an intelligent management server implemented in Java. This article concludes that the Java technology is adequate for writing mobile intelligent management agents. This author agrees and reiterates the fact that this technology should have an important role on management delegation (especially code delegation) due to its wide support and deployment.

Finally, a note on the recommended reference [75]: it discusses the usage of the Java technology as a mean of enabling the adaptation of the TMN framework to standard computing network services management.

4.5 Web-based Management

This type of technology is gaining much endorsement and effective deployment on commercial management systems. Its main strengths are the use of the most widely available application transfer protocol (HTTP), low consumption of device resources and rapid development of management applications using several powerful web interfaces.

Despite this hype around this technology, this author feels that its major usage on the long run will be mainly for implementation of management applications interfaces and initial remote device configuration when no management entities are still available, as endorsed by [82]. It should not be used has the structural technology behind a network services management framework.

The DTMF has recently defined a Web-Based Enterprise Management framework (WEBM) on [172] for deployment of management of local enterprise network services in the context of computing environments. This approach re-utilizes existent web technologies. The management data model uses the language and methodology specified by the CIM schema. The management data objects are then encoded into XML elements written in Document Type Definition using the xmlCIM Encoding Specification. The communications model uses a mapping methodology that maps CIM operations into HTTP operations.

References [67,72,73,109,134,159] present proposals for integration of web technologies (including Java) into the standard INMF. The proposal on [67] uses a complete event-driven[i] management architecture named WebView/98 and integrates Java and Web technologies with traditional SNMP and CMIP management protocols. The paper [72] presented a much more simple but useful and effective technology for implementation of a communications service between a management application and a management application's remote console or between management applications. References [73] and [109] present similar concepts for integration of web technologies with the INMF, although the specific usage of each technology and the integration methodology is not always identical. Nevertheless, these are reference works on this area. Finally, [134] presents a compact but interesting discussing of integration of Java technologies into SNMP management using SNMP Servlets, enabling the mapping of web-based management procedures into SNMP operations. Finally, document [159] proposes a mapping gateway between INMF MIBs and XML schema, enabling this way the use of web-based technologies to interact directly with SNMP management entities.

Some more bold proposals can be found on references [40,117,144], which define, each one of them, a complete management framework or management environment only using web-based technologies without integration with other traditional management frameworks. The first paper is an earlier classic resource on this type of approach, and defines a scalable architecture named Web Integrated Network for Distributed Management Including Logic, aka WINDMIL. This is a very simple proposal that could only be applied effectively for low level remote monitorization and configuration of network devices and distributed applications. The design behind the management system defined on [117] is similar to that of [159] but with no integration of traditional management communications protocols. Reference [144] presents only the conceptual architecture and information model driven only by web technologies and independent of any chosen underlying management communications and functional model.

4.6 Active Management

This is not a recent trend on network management technologies but has gained renovated attentions in the field due to developments on the research of mobile and intelligent agents, although a wide myriad of network management technologies and mechanisms could be used as the basis for deployment of this type of management.

Active, proactive or reactive techniques for network management all represent the same concept: the use of advanced techniques of monitorization to predict network, network services or distributed applications future operational status so the management system can react in advance by issuing the adequate management procedures that should drive the network, network services or distributed applications to a desirable future operational status.

The most important technologies supporting this type of management have to do with predictive monitorization algorithms integrating advanced artificial intelligence processes.

Reference [38] defends the integration of a proactive management module between the processes on a SNMP agent implementing monitorization MIBs and the SNMP manager receiving the predictive monitorization results (which should be very different than original MIB values gathered). The proposal of [58] is similar to this but is more complete and further elaborates on how to implement the predictive module. Also, the set of rules of the interface of this predictive module is named Active Virtual Network Management Protocol. On [70,77,85], an extra technology (management entities mobility) is integrated allowing the predictive modules to migrate on the management system and dynamically modifying themselves, depending on the predicted management requirements.

For an in depth analysis on the complexity of attaining feasible prediction results from network monitorization, please consult [115], which presents the reader with an insightful description of advanced

[i] This proposal extended the definition of SNMP trap events for support of real-time notification of events between management entities and management entities modules. This feature justified the chosen terminology by the article's authors.

algorithms, based on the Kolmogorov Complexity theory, to be deployed on active network management frameworks.

Finally, document [98] provides similar concepts of active management specifically applied to QoS management and predictive resources allocation.

This page was intentionally left blank.