Chapter 1

I. INTRODUCTION

Section I.1 – Background and motivations

In the last decade the business world witnessed the rapid development of Internet and IP networks in private and corporate areas. The wide acceptance of IP originated from its unparalleled ability to provide ubiquitous access and low prices regardless of underlying networking technologies. Moreover, based on the existing best-effort IP transport service, new application services can be offered on a global scale by almost everyone, simply by connecting a new web server to the Internet. Today IP is considered as unique glue to bridge diverse application/user requirements with broadband transfer capability. Various research initiatives such as NGI (Next Generation Internet), CANARIE, Internet2, etc., are progressing to provide unlimited bandwidth for Internet users. In parallel, based on the conventional Internet architecture, IETF (Internet Engineering Task Force) is performing a "bottom-up" development of Internet protocols and techniques, to fulfil upcoming requirements from applications, users, and providers.

However developing and deploying new network services, i.e. services which operate on the IP layer, through best practice and standardisation is too slow, and cannot match the steps in which requirements of various applications, e.g. multimedia multiparty communication, are growing. Examples of such services are signalling for quality of service (QoS), reliable multicast or Web Proxies/Caches/Switches/Filters. Similar to the intelligent network (IN) architecture in the PSTN world, the current Internet architecture needs to be enhanced in order to allow for a more rapid introduction of such services.

Programmable and active networks have been proposed as a solution for the fast and flexible deployment of new network services. The basic idea is to enable third parties (end users, operators, and service providers) to inject application-specific services (in the form of code) into the network. Applications are thus able to utilise these services to obtain required network support in terms of, e.g. performance, that is now becoming network-aware. As such, active networks allow dynamic injection of code as a promising way to realise application-specific service logic, or perform dynamic service provision on demand. But the dynamic injection of code can only be acceptable by network providers if it does not compromise the integrity, the performance and /or the security of networks. Therefore, viable architectures for active networks have to be carefully engineered to achieve suitable trade-offs among flexibility, performance, security and manageability.

Network management, either of telecommunication or data networks, has long been argued along the manager-agent model [Aidarous97] and deals with three fundamental aspects: a) *functionality* grouped according to five areas, namely, Fault, Configuration, Accounting, Performance and Security (FCAPS) [ITU00], b) *information modelling* by which network and network element resources are identified and abstracted in a way that underpins specific operational semantics, and c) the *communication method* among managers and agents.

According to [T101], network management consists on the execution of a set of functions required to control, plan, assign, deploy, coordinate and monitor telecommunication network resources, including performance functions such as planning the initial network, assign frequencies, route traffic oriented to support load balancing, cryptographic key distribution authorisation, configuration management, fault management, security management, performance management and accounting management.

The complexity of network management tasks lie in the fact that the managed components have evolved from isolated, homogeneous, controllable set of systems to a large, heterogeneous, distributed communication environment.

Being faced with such challenges several standards have been specified with the goal of supporting cross-system, multivendor networks. Management communication frameworks like the Simple Network Management Protocol (SNMP) [Case90] and Common Management Information Protocol (CMIP) [ITU97] have dominated from the early days of management. Recently, the emergence of advanced technologies, like the Common Object Request Broker Architecture (CORBA) [OMG02] and the Remote Method Invocation (RMI) [SunJAVAe] address the distribution of software environments and the interoperability of systems. These technologies ease the development of more open, interoperable, flexible and scalable management architectures.

However, there are still steps to be taken when it comes to the management of networks, as active and programmable networks, that are continuously changing their functionality and consequently their expectations from a management platform [Dimopoulou03].

More recently, policy-based networking has attracted significant industry interest [IPHighway]. Presently, it is promoted by several network equipment vendors in the form of fora like DMTF [DMTF] or is standardised within the IETF Policy working group [IETFPol]. Policy-Based Network Management (PBNM) opened a new window of opportunity to operators as it enables them to homogeneously perform their network management tasks, raise the level of interoperability across different vendors' equipment thereby creating a new range of different customisable service products.

PBNM technology is particularly suited for handling the particular characteristics of active and programmable networks. More specifically, policies are particularly suited for delegating management responsibility, essential to enable the customisability of network resources. Also, the device-independent property of policies is optimum for the management of heterogeneous network technologies. In addition, policies permit a more automated and distributed approach to management, taking decisions based on locally available information according to a set of rules [Prnjat02].

These advantageous properties of PBNM technologies for managing active and programmable networks have inclined us to use this technology in our proposed solution. In the following sub-sections we provide detailed descriptions of the two main technologies involved in this thesis: active and programmable network technology and policy-based network management technology.

1st Introduction to active and programmable networks

The rapid deployment and customisation of the offered services lead to the introduction of programmability in the network elements. This first occurred in the field of telecommunications, with the Intelligent Networks (IN) [ITU92] and the Advanced Intelligent Networks [Bell] and spread to the data communication community with the emerge of open signalling and active networks. These advanced are driven by a service-oriented market that needs granularity, openness and reduced time-to-market.

There is a misunderstanding and usually confusion when people speak about active and programmable networks. The term programmable networks is used widely by the Opensig community [Opensig] to characterise networks that are build on the principles they promote. The networking research community has realised for sometime now the need for more flexibility and dynamically customisable networks. Therefore the change of the one-dimensional networking model based on the communication model (realised by packet header processing and forwarding), to the two-dimensional one with the addition of the computational model seemed the next step to network evolution. A programmable network realises this, by allowing a third party to customise and process the packets that pass from the network interface by calling open interfaces that reconfigure the node or even execute programs on that node. However, these programs are predefined and limited, in the sense of interfaces, and thus capabilities; they are initiated via predefined interfaces or called with specific parameters and bring the node to deterministic states.

On the other hand, active networks are a new generation of networks based on a software-intensive network architecture in which applications are able to change the network behaviour or tailor the infrastructure to their immediate needs. This allows them to be flexible and extensible at run-time so that they can accommodate the rapid evolution of new technologies and the deployment of new sophisticated services. At the heart of the active network is the active packet, which provides the basis for describing, provisioning, or tailoring network resources to achieve application requirements.

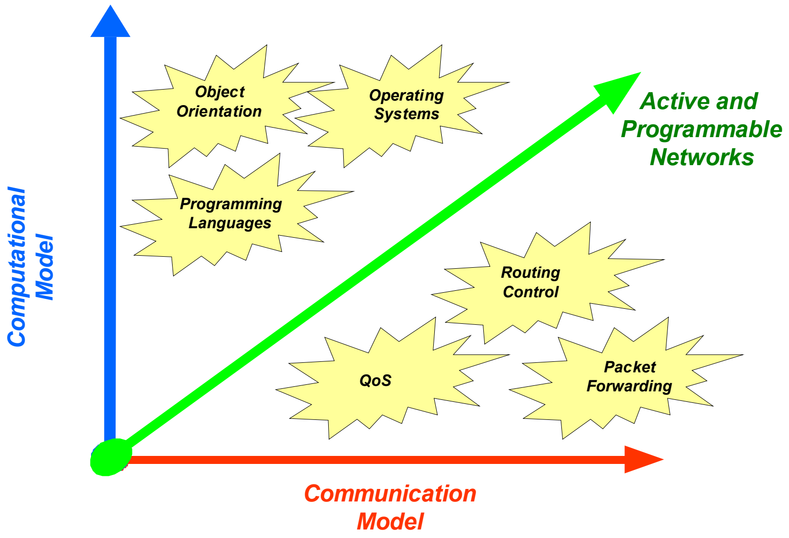

Figure 1 - 1. Active and Programmable Networks Problem Space

The two dimensional model depicted in Figure 1 - 1, shows the networking paradigm shift that we are witnessing from the flat communication model to a new two-dimensional space that is composed of the mix of communication and computational models. Traditional parts of both models, like packet header processing and forwarding, Quality of Service in the communication model interact with the technologies such as programming languages, distributed programming of the computational model. The result is that network elements (e.g. routers, firewalls, switches etc) that were exclusively developed with closed proprietary interfaces are now considered as a place where advanced customised computation may take place.

One major issue in the large-scale deployment of active networks would be its interoperability with legacy network nodes. An interesting feature is that there would be no need to overhaul the existing infrastructure. Active nodes can co-exist with legacy nodes by *tunnelling* through them using some of the existing approaches [ANEP], [Decasper98].

As identified before, two different schools of thoughts are dealing with this new problem space, namely the Opensig [Opensig] and DARPA [anets]. During the last years there is a significant involvement of the international community as active networks gain momentum, and several research efforts are published in conferences and journals while in parallel the number of projects in this domain has increased exponentially.

A Standards and ongoing research activities

a Open Signalling

i The IEEE P1520

The original motivation behind Opensig networks has been the observation that monolithic and complex control architectures may be restructured according to a minimal set of layers where the services residing in each layer are accessible through open interfaces thus providing the basis for service creation (composition). Eventually, a number of results out of the Opensig community were formalised by the IEEE Project 1520 standards initiative for programmable network interfaces and its corresponding reference model [Biswas98]. The IEEE P1520 Reference Model (RM) provides a general framework for mapping programming interfaces and operations of networks, over any given networking technology.

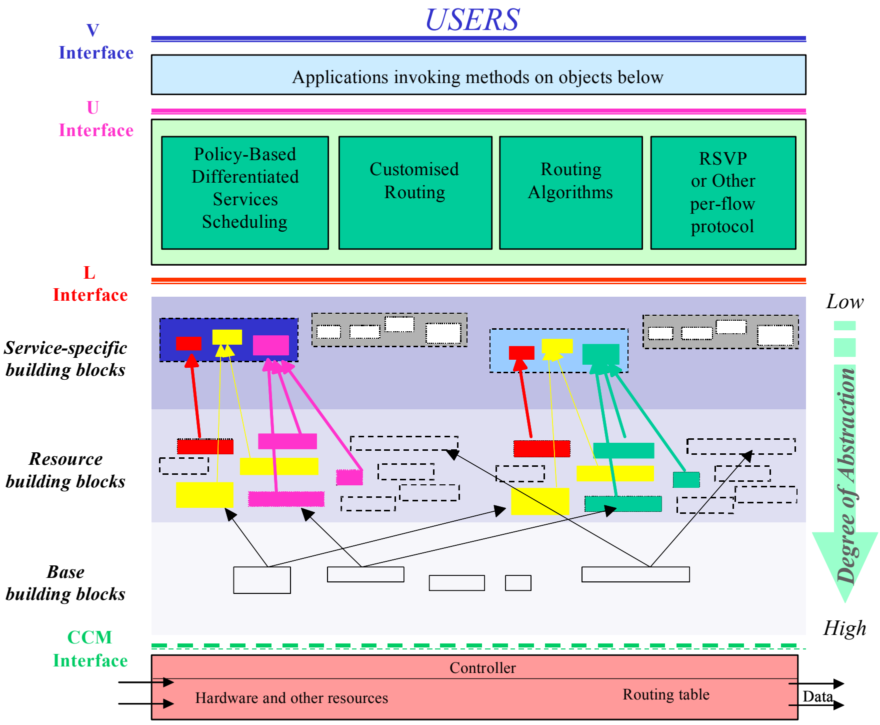

Figure 1 - 2. The P1520 Reference Model and the L-Interface abstraction model

The IEEE P1520 RM, depicted in Figure 1 - 2, defines the following four interfaces:

CCM interface: The connection control and management interface is a collection of protocols that enable the exchange of state and control information at a very low level between the Network Element (NE) and an external agent.

L-interface: It defines an Application Program Interface (API) that consists of methods for manipulating local network resources abstracted as objects. This abstraction isolates upper layers from hardware dependencies or other proprietary interfaces.

U-interface: It mainly provides an API that deals with connection set-up issues. The U-interface isolates the diversity of connection set-up requests from the actual algorithms that implement them.

V-interface: It provides a rich set of APIs to write highly customised software often in the form of value-added services.

CCM and L-interfaces fall under the category of NE interfaces, whereas U- and V-interfaces constitute network-wide interfaces.

Initial efforts through the ATM Sub-Working group (P1520.2), focused on telecommunication networks based on ATM and introduced programmability in the control plane [P1520]. Later, The IP Sub-working group extended these principles to IP networks and routers. Figure 1 - 2 also suggests a possible mapping of the P1520 RM to IP routers. However, their efforts focus on creating a generalised framework for designing interfaces not just for routers but also for any NE the core functionality of which is forwarding of traffic, e.g. switch, gateway etc [Biswas00].

ii The IETF ForCES

The Opensig community has long advocated the benefits of a clear distinction between control and transport plane. Recently, a working group of IETF, called ForCES (Forwarding and Control Element Separation) was formed with a similar objective to that of P1520, namely, “by defining a set of standard mechanisms for control and forwarding separation, ForCES will enable rapid innovation in both the control and forwarding planes. A standard separation mechanism allows the control and forwarding planes to innovate in parallel while maintaining interoperability” [IETFForces], [IETFForces02].

According to [Vicente00], the NE is a collection of components of two types: control elements (CE) and forwarding elements (FE) operating in the control and forwarding (transport) plane, respectively. CEs host control functionality like routing and signalling protocols, whereas FEs perform operations on packets, like header processing, metering, scheduling, etc. when passing through them. CEs and FEs may be interconnected with each other in every possible combination (CE-CE, CE-FE and FE-FE) thus forming arbitrary types of logical topologies (see Figure 1 - 3). Every distinct combination defines a reference point, namely, Fr, Fp and Fi. Each one of these reference points may define a protocol or a collection thereof, but ForCES protocol is only defined for the Fp reference point.

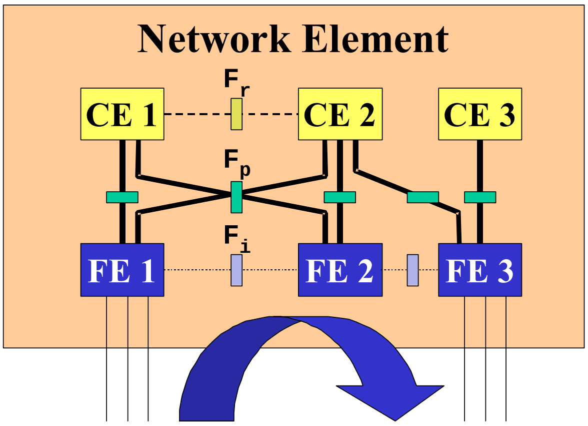

Figure 1 - 3. ForCES Architectural Representation of NE

However, FEs do not represent the smallest degree of granularity of the NE functionality. Furthermore, as they implement the ForCES protocol they must facilitate CEs to control them in terms of abstracting their capabilities, which, in turn may be accessed by the CEs. It is at this point that the ForCES group faced a similar challenge as the IP working group in P1520 which they formulated it as follows: Since FEs may manifest varying functionality in participating in the ForCES NE, "the implication is that CEs can make only minimal assumptions about the functionality provided by its FEs" [Yang03]. As a result, CEs must first discover the capabilities of the FEs before they can actually control them.

The solution they suggest is captured in the form of an FE Model [Yang03], while two of its requirements that must satisfy pertain to the problem of an extensible standard. The first mandates that the FE model should provide the means to describe existing, new or vendor specific logical functions found in the FEs, while the latter demands to describe the order in which these logical functions are applied in the FE [Khosravi03].

In the ForCES FE model, they use a similar approach to the building block approach of the P1520.3 working group, by encapsulating distinct logical functions by means of an entity called, FE block. When this FE block is treated outside the context of a logical function, it becomes equivalent of the base building blocks. When someone looks what is inside every FE block then it becomes a resource building block. Similarly, FE blocks eventually are expected to form an FE block library – in principle extensible-, which will be part of the standard and the basis for creating complex NE behaviours, although dynamic extensions thereof may be possible.

A type of model like the FE model is useful when CEs attempt to configure and control FEs. ForCES has identified three levels of control and configuration, namely, static FE, dynamic FE, and dynamic extensible FE control and configuration. The first assumes that the structure of the FE is already known and fixed, the second one allows the CE to discover and configure the structure of the FE although selecting from a fixed FE block library, whereas the third one is the most powerful that allows CEs to download additional functionality, namely FE blocks, onto FEs at runtime. Currently ForCES is mainly, focusing on the first level of control and configuration.

b DARPA Active Networks

DARPA (Defense Advanced Research Projects Agency) promoted active networks by funding a program for their research [anets]. Around fifty projects have been included in this program. It had the goal of producing a new networking platform, flexible and extensible at runtime to accommodate the rapid evolution and deployment of networking technologies and to provide the increasingly sophisticated services demanded by defence applications.

Active Networks transform the store-and-forward network into store-compute-and-forward network. The innovation here is that packets are no longer passive but rather active in the sense that they carry executable code together with their data payload. This code is dispatched and executed at designated (active) nodes performing operations on the packet data as well as changing the current state of the node to be found by the packets that follow. In this context, two approaches can be identified based on whether programs and data are carried discretely, namely within separate packets (out-of-band) or in an integrated manner, i.e. in-band.

In the discrete case, the job of injecting code into the node and the job of processing packets are separated. The user or network operator first injects his customised code into the routers along a path. Then the data packet arrives, its header is examined and the appropriate pre-installed code is loaded to operate on its contents [Wetherall98], [Decasper99]. Separate mechanisms for loading and executing may be required for the control thereof. This separation enables network operators to dynamically download code to extend node's capabilities, which in turn they become available to customers through execution.

At the other extreme lies the integrated approach where code and data are carried by the same packet [ITU92]. In this context, when a packet arrives at a node, code and data are separated, and the code is loaded to operate on the packet's data or change the state of the node. A hybrid approach has also been proposed [Alexander98].

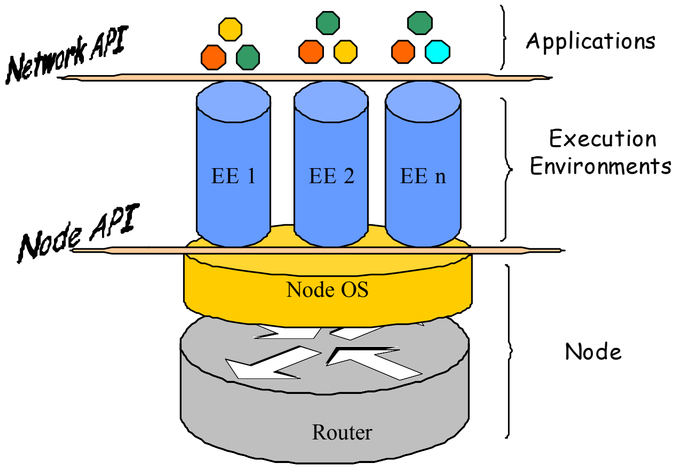

Figure 1 - 4. The Active Node Architecture

Active networks have also proposed their own reference architecture model [Calvert99] depicted in Figure 1 - 4. According to it, an active network is a mixture of active and legacy (non-active) nodes. The active nodes run the node operating system (NodeOS) –not necessarily the same- while a number of Execution Environments (EE) coexist at the same node. Finally a number of active applications (AA) make use of services offered by the EEs.

The NodeOS undertakes the task of simultaneously supporting multiple EEs. Accordingly, its major functionality is to provide isolation among EEs through resource allocation and control mechanisms, and providing security mechanisms to protect EEs from each other. It may also provide other basic facilities like caching or code distribution that EEs may use to build higher abstractions to be presented to their AAs. All these capabilities are encapsulated by the Node interface through which EEs interact with the NodeOS. This is the minimal fixed point at which interoperability is achieved [Peterson01].

In contrast, EEs implement a very broad definition of a Network API ranging from programming languages to virtual machines like the Spanner VM in Smart Packets and bytecodes [Schwartz99], to static APIs in the form of a simple list of fixed-size parameters etc [Calvert98]. To this end, EE takes the form of a middleware toolkit for creating, composing and deploying services.

Finally, the active networks reference architecture [Calvert99] is designed for simultaneously supporting a multiplicity of EEs at a node. Furthermore, only EEs of the same type are allowed to communicate with each other, whereas EEs of different type are kept isolated from each other.

c Other initiatives

Although the Opensig and DARPA communities are the most relevant ones in the field of active and programmable networks, there are other efforts also in this field.

Working groups 6.6 [IFIPb] and 6.7 [IFIPa] within the International Federation for Information Processing (IFIP) aim to study new network management techniques (including policy based management, active networks and mobile agents) for the management of future networks and study current issues related to the development of intelligent capabilities in networks (including Intelligent Agents, Active Networks, Programmable Networks, Hybrid Networks, Configurable Architectures for software and hardware, Dependable Reconfigurable Networks, Mobility Management, QoS Management and Network Integration issues) respectively.

The Eurescom project 844 (P844) [Eurescom] is a short project with the goal of realising a strategic study in the impacts of active networks. The result of the project stated: "...the application of active network ideas is not only useful but also necessary in creating wide area systems incorporating the studied basic services".

The Information Society Technologies Programme (IST) [IST] of the European Union is also funding several active networks research projects such as FAIN, Android and others.

2nd Introduction to Policy-based Network Management

Policy-based Network Management (PBNM) automates network infrastructure control by storing policies on a centralised server where they can be pushed out to the networking infrastructure. Policies are abstracted to apply across a variety of different devices so there is no need to create separate rules for each policy client. At the device level, policies are implemented by means of an "If/Then" proposition. That is, if certain conditions are present, then specific actions are to be taken. An "If " condition can be a time of day, a type of traffic, an IP address, a person, a group, or combinations of these. A specific action might request the configuration of priority tagging or set security encryption at a certain level. Other possibilities are actions related to access and load balancing, and more sophisticated traffic-shaping.

PBNM defines two main models for policy management; these are outsourcing and provisioning.

The outsourcing model assumes that there is a signalling request from the managed device that must be authorised based on policy criteria. Signalling requests are typically associated with an end-to-end signalling protocol (such as RSVP, MPLS-LDP, Multicast Join ICMP, etc.) The outsourcing model is sometimes referred to as "Pull" mode, or "reactive" mode, because on the one hand, the managed device pulls policy decisions from the PBNM system, and on the other hand, the PBNM reacts to those events.

The provisioning model is almost the reverse of the outsourcing model. It is the PBNM system the one that predicts future configuration needs, and proactively pre-provisions for them ahead of time. Rather than responding to device requests, the PBNM prepares and "pushes" configuration information to the device, as a result of an external event, such as change of applicable policy, time of day, expiration of account quota, or as a result of third party signalling. The provisioning mode is most commonly used for controlling network policy for non-signalled protocols, such as DiffServ, or configuring devices for particular services (such as VPNs or VoIP).

A Standards and Working groups

The use of policies for network management has recently been introduced in the Internet community. However, for the deployment of Policy Based Network Management systems in the Internet, a standardisation process is required, to ensure the interoperability between equipment from different vendors and PBNM systems from different developers.

Both the Internet Engineering Task Force (IETF) [IETF] and the Distributed Management Task Force (DMTF) [DMTF] are currently working for the definition of standards for Policy Based Network Management. The DMTF is mainly focused on the representation of policies and the specification of a corresponding information model and schema. The IETF is also working in that field, in co-operation with DMTF, while also trying to define a general framework for a PBNM system, as well as a protocol that could be used for implementing a PBNM system.

a DMTF work on Policy Based Network Management

The DMTF has defined the Common Information Model (CIM) [DMTFCIM] management schema, which consists of an object-oriented model for the representation of the information that will be stored in the directory of a Directory Enabled Network (DEN) [DMTFDEN]. The CIM has been the starting point for the specification of the Policy Core Information Model by the IETF.

The CIM specification is in a stable state. Newer work related to policies is carried in the IETF policy framework workgroup in coordination with the DMTF.

b IETF work on Policy Based Network Management

There are several groups within the IETF where activity related to Policy Based Network Management is taking place. The IETF working groups that are more related to Policy Based Network Management are the Policy

Framework (policy) workgroup [IETFPol], and the Resource Allocation Protocol (rap) [IETFRAP] workgroup.

The target of the Policy workgroup is first, the specification of a framework for Policy Based Network Management (see Figure 1 - 5). Second, the definition of the Policy Core Information Model (PCIM) [Moore01], [Moore03] for the representation of generic policy data and finally, the extension of the PCIM to support policies related to QoS traffic management [Snir03], [Moore03b].

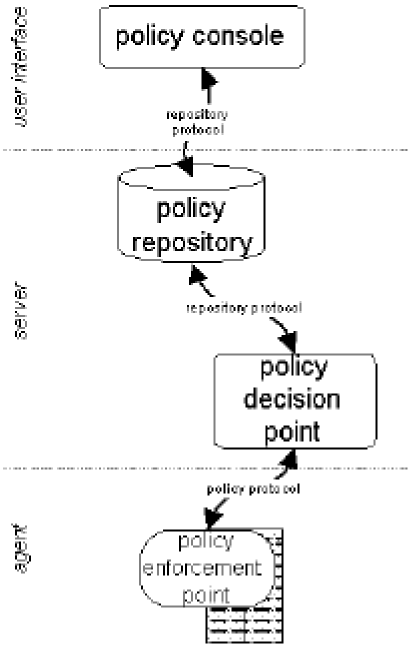

Figure 1 - 5. Policy-based network management framework

The proposed framework consisted of four elements. The policy console offers a user interface for introducing policies within the PBNM system. These policies are stored in the policy repository from where they are retrieved by the Policy Decision Point (PDP) to decide when they should be enforced. Finally, the Policy Enforcement Point (PEP) is in charge of configuring the managed device accordingly when policy conditions are met.

The Resource Allocation Protocol working group has defined the COPS protocol [Durham00a], for the communication between a policy decision entity (Policy Decision Point) and the device where the policy is enforced (Policy Enforcement Point). Additionally, this workgroup is working towards the definition of general-purpose objects that facilitate the manipulation of policies and provisioned objects available through COPS. Finally, the RAP working group has also work on the application of COPS over the Resource Reservation Protocol (RSVP) [Durham00b].

c Other groups

Other working groups in policy based network management are the IETF Snmpconf working group [IETFSNMPConfa], which is defining objects that enable policy-based configuration management of SNMP infrastructures [IETFSNMPConfb] and the Foundation for Intelligent Physical Agents (FIPA) [FIPA], which are starting policy work related to Intelligent Agents [FIPAa].

B Policy Based Network Management Tools

In this subsection we develop a brief description and comparison of those policy-based network management tools with more relevance on the market. This analysis is made in order to assess if any of these tools could be used as starting point for the work targeted in this thesis.

The first interesting particularity is that in all commercial PBNM products analysed, policy definition is done through a graphical user interface. First, the administrator selects the device or group of devices to which the policy will be applied. Then, he can select in a menu one or more supported condition types and supply the concrete parameters for the selected conditions. In a new menu, he can select one or more supported actions. Usually, the PBNM tool identifies the selected devices, to check their capabilities and list at the user menus only those conditions and actions supported by the devices.

The conditions supported by the commercial tools mainly fall into two large categories: time-based conditions and packet-based conditions (based on the packet header).

The actions that can be applied when the conditions of a policy are satisfied include the configuration of the queuing mechanism on the router, traffic colouring, denial of service to the specific flow, prioritisation of traffic, etc.

The format of the policy rules, that is the format in which the policies are stored in the directory and the format in which policies are transferred to the target devices, is different in each product. However, some of the tools use the DMTF's CIM model for storing the rules in the directory.

The configuration of the network devices when policies are enforced is done with a variety of protocols. Nonetheless, most of the PBNM developers use the COPS protocol for communication with the devices. If a device does not support COPS, a COPS proxy agent is used for the configuration. Also SNMP is often used for device configuration. Other platforms use also CLI (Command-line interface) commands.

In the next table we summarise these and other properties of the PBNM tools analysed.

	Database support	Configuration Protocols	Device support	Policy conditions	Advantages	Disadvantages
Allot	LDAP	COPS, CLI	Allot, Cisco	Layer 1, 3, 4, 5+, time based	Active feedback mechanism	Proprietary.
Cisco	Flat-file	CLI	Cisco, Hitachi, Lucent	Layer 3, 4	Can import the topology from Cisco application. Define policies on a per-interface basis	No CIM or LDAP. No COPS support yet. No time-based conditions
Extreme	LDAP	COPS	Cisco and other	Layer 1, 2 (partially), 3 (partially), 4	Supports integration with a WinNT environment	No device discovery. No time-based conditions.
HP	?	COPS	HP switches, Cisco, Intel, Nortel routers	Layer 1, 2 (no VLAN) 3, 4, 5+ and time based	RSVP capabilities, and use of COPS for policy provision	No LDAP support. No device discovery. No queuing configuration.
IPHighway	?	COPS and CLI	Cisco routers and any COPS device	Layer 1, 3, 4 and time based	RSVP capabilities	Focused on COPS devices
Lucent	LDAP (CIM schema)	LDAP, SNMP, COPS (but not used)	Cisco LAN routers, Lucent switches.	Layer 3,4, time based conditions	Extended use of LDAP. Can define policies based on particular hosts and users, can translate a user's MAC address to an IP-based policy.	Limited set of conditions and actions. No COPS application.
Nortel	Oracle database	SNMP, CLI	Cisco and Nortel	Layers 1-4, time-based	No special advantage	No LDAP and COPS
Orchestream	Oracle database	SNMP, CLI and TACACS+	Cisco, Lucent, Xedia	Layer 3, 4 and time based	Network topology discovery. Special emphasis on Diffserv	No COPS. No support for Layer 2 conditions.
Spectrum	LDAP	CLI,SNMP	Cabletron switches and routers	Layers 1-4 and time-based	Largest range of conditions. Significant IP and IPX support. Latest CIM specifications. Topology aware	Lack of multidevice management. No COPS support yet.

Table 1 - 1. Summary of existing policy based management tools properties.

All PBNM tools analysed presented a number of drawbacks for their use in the present proposal. Hereafter, we proceed to enumerate these drawbacks.

First, the existing policy based management tools are commercial products and consequently, are targeted at the management of existing business networks. Thereby, any of the analysed tools would be able to cope with the specific characteristics of active and programmable networks. This is due to the fact that the primary concern of vendors is the delivery of a functional PBNM system that can be used by customers, without necessarily focusing on specific standards or trying to provide an open solution. Moreover, we can observe that almost all the developers of PBNM systems are also manufacturers of network devices, whose main aim is to provide a good management platform for their own products.

Second, the conditions and actions available in the PBNM tools analysed are scant. Conditions belong just to two main categories: time-based and packet-based. Moreover, it is not possible to define new policies based on the status of an active service, the node, the network, etc.

Another drawback is the fact that each tool supports a specific set of hardware devices and the introduction of a new device is not feasible, if it does not belong to the list of supported devices.

Summarising, all analysed tools present several deficiencies for being used in the management of active and programmable networks, and in addition, they are closed in the sense that they do not permit the modification of their code to overcome these limitations. Therefore, none of the analysed tools was considered adequate for its use in the current proposal, so we will design and implement our own policy-based network management tools.

Section I.2 – Objectives of this Thesis

Once the motivations and contexts for this research work have been explained, we are going to describe the objectives pursued.

The main objective is:

> *"The proposal of a management framework intended to the management of heterogeneous active, programmable and passive networks supported on the policy-based network management paradigm and facilities of the active networking technology. We have named this framework MANBoP that stands for Management of Active Networks Based on Policies".*

This management framework must exhibit the following attributes or characteristics that can be considered as secondary objectives:

i) The proposed framework must be flexible. It must be able to deal with different underlying devices of different types, different services, etc. and it must do it efficiently. That is, on the one hand, it must be able to manage a set of active network specific issues like the efficient management of code mobility and management of computational resources like CPU, memory, etc. On the other hand, it must take advantage of the facilities offered by underlying devices to enhance the management mechanism. Furthermore, it must support its instantiation at different management levels to permit the creation of different management infrastructures. This pursues the goal of permitting to network operators the simple creation of the management infrastructure that best suits their needs based on their business objectives, managed network topology, number of users, etc.

ii) The management framework must be extensible at run-time. More specifically the management framework must support first, the dynamic extension of management functionality. That is, the management functionality must be modified when required to handle new functionality installed in the active network. Second, it must support also the dynamic addition and removal of managed devices in the managed network. The target of this second aspect is to ease and promote the evolution of legacy passive networks into active and programmable networks by progressively substituting network elements.

iii) The proposed solution must support the delegation of management functionality to users (e.g. service providers, consumers). The target of this objective is on the one hand, to allow service providers and consumers to have better control over the service as well as over the network resources the service is using. On the other hand, to save network administration costs to network operators.

iv) The management framework designed and developed must be scalable so that it can cope with potential increments in the number of managed devices and user requests.

v) The management framework must facilitate to the maximum extent the interworking with other systems. The goal of this objective is to simplify the integration of the MANBoP framework with other applications and services.

vi) The proposed MANBoP framework must be fully portable to any kind of machine as long as it offers the minimum computational and communication resources required by the framework.

The fulfilment of these objectives will be assessed in Chapter Six. Moreover, these objectives will be used as functional criteria that will guide the evaluation process.

The main objectives of this Thesis have been summarised in the following figure:

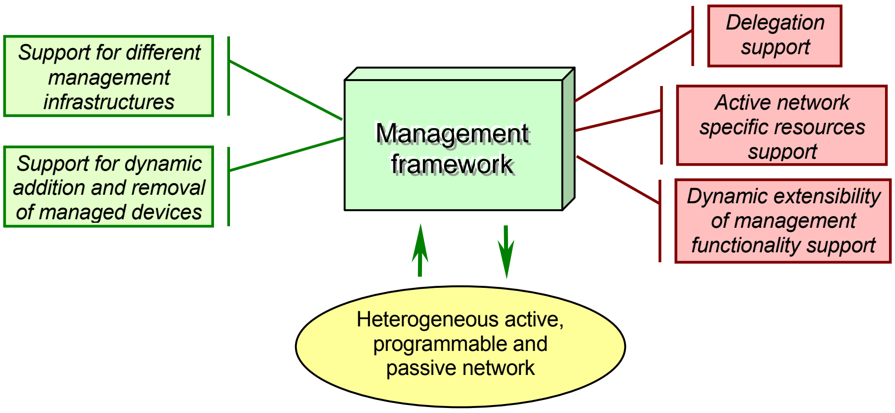
Support for different management infrastructures
Support for dynamic addition and removal of managed devices
Management framework
Delegation support
Active network specific resources support
Dynamic extensibility of management functionality support
Heterogeneous active, programmable and passive network

Figure 1 - 6. MANBoP objectives

Section I.3 – Document structure

In the present chapter we have provided a detailed overview of the work background. We have first described the motivations over which the proposal is based as well as the main technologies involved. Then, we have listed the objectives of the thesis.

The following chapters detail different aspects of the work developed. More specifically, the second chapter analyses the requirements set over the management framework proposed. These requirements have been split in two groups. The first one contains all requirements that active and programmable networks impose over the management system, while the second one contains other requirements derived from the project objectives or others.

Chapter Three provides an overview over the most relevant projects that have explored similar fields to those covered in this thesis. More specifically, the projects analysed have been grouped in those that propose non policy-based management architectures for active and programmable networks and those that suggest policy-based management architectures for active and programmable networks.

Chapter Four provides en exhaustive description of the MANBoP design. The design description is supported by several UML diagrams and figures that aim to ease its comprehension. First, we describe the functionality supported by the design in the form of use cases. Then, we include a detailed description of how this functionality is developed by the different components designed in the framework.

Chapter Five describes the proof-of-concepts implementation of the proposed framework as well as the Information Model followed. At the end of this chapter we propose a couple of scenarios that will be used for evaluating the proof-of-concepts implementation developed.

Chapter Six assesses the proof-of-concepts implementation of the framework based on the results obtained from running the two scenarios described in the previous chapter. All recompiled data is provided and commented, often in the form of figures to ease its assimilation. We elaborate around the reasons for these results and compare them with results from other projects when possible.

Finally, Chapter Seven summarises the main outputs from the work realised and comments its strengths and weaknesses. Additionally, it suggests future work that can be developed to enhance the management framework proposed.

At the end of the document, we include four appendixes that contain further information around the MANBoP framework. Particularly, appendix A provides the definition in Interface Definition Language (IDL) of all MANBoP components' interfaces elaborated for the proof-of-concepts implementation.

Appendix B describes the structure and information contained within the files introduced in the bootstrapping of the MANBoP framework. These files provide the initial configuration information for the framework.

Appendix C details the information and structure of XML policies and the corresponding XML Schemas specified for the proof-of-concepts implementation.

Appendix D describes a number of tools that have been implemented to ease the development of the proof-of-concepts in one hand, and to facilitate the interactions with the framework in the other hand.