

A Framework for Policy-Based Management of QoS Aware IP Networks



[Metadata, citation and similar papers at core.ac.uk](http://core.ac.uk)

Provided by Archivio della ricerca - Università degli studi di Napoli Federico II

G. Ventre

¹NETIKOS - via Matteucci, 56100 Pisa
piorgio.giorgio.cremonese@netikos.com

²DIS -- Dipartimento di Informatica e Sistemistica, Università di Napoli Federico II
Via Claudio 21, 80125, Napoli, ITALY
{mesposit, spromano, giorgio}@unina.it

³ICA -DSC- EPFL - CH-1015 Lausanne, Switzerland
silvia.giordano@epfl.ch,

Abstract. In today's Internet, Policy-Based Network Management is gaining more and more proselytes. Its appeal is due to the given opportunity of a standard and consistent way for network configuration, independently of the underlying architecture and Quality of Service (QoS) provisioning model assumptions. The event-driven paradigm, well established in the general-purpose programmers world, through the Policy-Based approach begins to play its role also in the field of network management. In this paper we describe a policy framework suited for dynamic network management in QoS-enabled IP networks. First, we design an object model conceived to represent policies in a network-independent fashion. Then, we describe a management and configuration system based on Common Open Policy Service (COPS). Finally, we show a system prototype pointing out the main features of a Differentiated Services network management and configuration based on Policy System Management.

1 Introduction

A policy is a set of rules or methods, representing an object behavior or a decision strategy to be applied in order to ultimate a particular goal. The Policy-Based Network Management is the application of these organizational policies in order to manage the networks. With this approach, the role of network management moves from passive network monitoring to active QoS (Quality of Service) and network service-level-agreement provisioning .

While this technology is powerful and alluring, it's also generally untested and unproven. Worse, this area still suffers from a lack of standards and for a lack of ad hoc use of existing ones. There are two key issues that are not yet totally addressed: first, how the vendors will access and control their hardware, and second, how these systems glean information about an organization's users and resources.

Our architecture, developed in the framework of the European IST project CADENUS, tries to address all those problems. First, we are developing a prototype that aims to test and validate the policy-based approach in a real DiffServ network. Secondly, we adopted a layered model. In this way, at the lower layer, we accomplished the devices configuration by employing a combination of CLI (command-level interface), COPS and LDAP. We feel that our work can be a step toward a standardized policy-based network management.

This document is specifically concerned with the definition of the processes that take place right after a new Service Level Specification (SLS) has been created as a consequence of the negotiation of a new service instance between, for example, an end-user and a Service Provider (SP). We are not focussing on the interactions that bring to the creation of an SLS, but simply assume that a new SLS has been provided by a Service Provider stemming from an even higher service level description (see [SLA]).

This document is organized in five sections. Next section illustrates the proposed architecture and the steps performed from an SLS to the final configuration of the devices. The Multiple-Layer, Policy-Based approach, with particular attention to the policies repositories used at each layer, is presented in Section 3. Section 4 expands on the Network Controller, which represents one of the main components of the overall architecture. Finally, Section 5 provides some concluding remarks, together with a discussion of future work.

2 A Framework for Automatic Configuration and Management of QoS-Aware Networks

Policy-Based Management has been thought to allow network configuration in the sphere of several applications, ranging from security and network engineering to monitoring and measurements. In this work, we will delve into the role of policies with respect to Quality of Service (QoS) needs in a QoS-aware network. In order to make network configuration and management an automatic task, independent of the specific devices implementing the network, our architecture is composed of three layers, each related to a different level of abstraction. More precisely, as depicted in Figure 1, the overall process starts from an abstract service description (contained inside an SLS), and comprises a number of intermediate steps, each needed in order to lower the level of abstraction, thus filling the gap between the human-oriented concept of a “service” and the device-specific configuration commands that eventually enforce the service itself. For each domain (i.e. Autonomous System --- AS) we have one functional block, named Resource Mediator (RM), that is in charge of managing the whole underlying network.

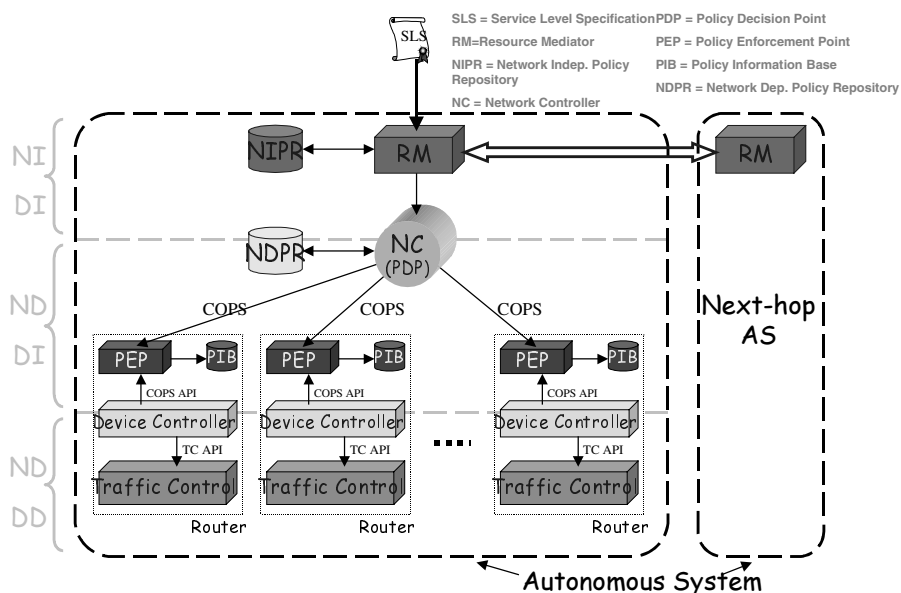


Fig. 1. Different layers of the Policy Architecture

The scenario we analyze is one in which, starting from an SLS instance, we go all the way down through the shown components in order to arrive at the network devices and appropriately configure them. Delving into the details of such a process, we identify the following steps:

1. A Resource Mediator takes an SLS and translates it into a coherent set of Network Independent Policy Rules (NIPR). As the name itself suggests, such rules are to be both network and device independent: they just are a well-structured representation of the information contained inside the SLS. The model we are thinking to adopt is inspired to the various proposals stemming from the Common Information Model [CIM] under standardization inside both the IETF and DMTF research communities [PCIM],[PCIME],[PQIM].
2. The Network Independent Policy Rules are then passed to a Network Controller (NC), which translates them on the basis of the specific network architecture adopted (MPLS, Diffserv, etc.). The translation process brings to a new set of rules, named Network Dependent Policy Rules (NDPR), that are stored inside an ad-hoc defined Policy Information Base (PIB). The NC also acts as a Policy Decision Point (PDP) [COPS], which exploits a protocol like COPS to send, based on the “provisioning” paradigm, policies to the underlying Policy Enforcement Points (PEPs).

3. Upon reception of a new policy, the PEP is now in charge of interacting with a Device Controller (DC), thus triggering the last level of translation, so to produce the necessary configuration commands needed to appropriately configure the traffic control modules (e.g. allocation and configuration of queues, conditioners, markers, filters, etc.) on the underlying network elements.

3 The Policy-Based and Multiple-Layers Approach

As we introduced in the previous sections, the innovative aspect of the CADENUS architecture is the policy-based and three-layers approach. The sequential steps performed by each layer aim to achieve the ultimate goal of setting up the network in an automatic fashion, without any human intervention. The SLS is an abstract service description, independent both of the network architecture (e.g. Diffserv, MPLS, ATM) and of the devices architecture (e.g. a CISCO router, a PC running Linux or FreeBSD). Yet, network-dependent and device-dependent information is still needed in order to configure and manage the network devices. For this purpose our architecture includes three databases containing the views of the requested QoS at different layers: the Network Independent Policy Repository, the Network Dependent Policy Repository, and the Vendor Dependent Policy Repository.

3.1 NIPR: A Repository for Network Independent Policies

The Network Independent Policy Repository (NIPR) is an archive located at the Resource Mediator (RM) level. When a new SLS arrives at the RM from the Service Provider, the RM translates it in a set of policy rules describing conditions and actions related to the requested service (still in a network-independent form) and stores it in the NIPR. This SLS must describe the single service instances in an unambiguous fashion.

The peculiarity of such an approach is that a NIPR, being at a high level of abstraction and then entirely network independent, can represent a common component (for every network architecture) containing the bundle of services to be enforced in the future. Anyway, as already stated, the semantic value of information stored in the NIPR, is not different than the one contained in the original SLS: the only added feature is the policy-based representation.

3.2 NDPR – A Repository for Network Dependent Policies

As we just explained, the Network Independent Policy Repository is a formal representation of the information contained inside an SLS. It contains, in standard format, the otherwise fuzzy definition of a *service*. In order to let such a definition become comprehensible to the lower network management devices, the need arises for a further level of translation. For this step, the Network Controller (see figure 1) goes a step further, by taking into consideration the specific network architecture that will support the deployment of the service. These network dependent policies are stored in the Network Dependent Policy Repository (NDPR). The NDPR contains

policies in a representation independent of the devices implementing the network. It introduces, in the service/flow description, rules deriving from the supported technology (e.g. Diffserv) without going in detail of devices characteristics and components. NDPR generation is performed by Network Controllers according to business rules defined for traffic classification. The mapping from NIPR (which is based on user/service requirements) to NDPR (based on network implementation) is local to each domain. Each NC uses some business rules for policy generation and policy distribution. Such a policy could, for example, lead to the marking (via DSCP field) of a packet, or dropping, remarking, delaying of out of profile packets.

A policy is defined by the instantiation of a filter object (condition) and an action object. A filter object identifies, for instance, source and destination, while an action object can include Classifier object, Meter object, Shaper object.

3.3 VDPR – A Repository for Vendor Dependent Policies

This layer works with a representation that can be understood and handled by devices, thus reflecting their specific characteristics. The policies defined at the previous layer are translated into device configuration policies. Information, which is vendor and device dependent, such as queues configuration and network interfaces, is added at this layer. The vendor dependency derives from the necessity to make rules according to the specific features of the managed device. The schema for the translation of PIBs changes with the device nature. Therefore, this translation is demanded to a dedicated component, named the *Device Controller*. In our case, this component has been implemented for Linux-based routers, exploiting the functionality made available by the Linux Traffic Control (TC) module.

4 The Network Controller

The Network Controller (NC) is the component responsible for network management and configuration. Each NC manages a homogeneous network, where “homogeneous” means that only one technology for QoS support is provided within the network. The NC role can be summarized as follows:

- it performs management and configuration based on requests coming from the RM;
- it provides the RM with data for updating local repositories (routing, resources);
- it provides input to devices for local Traffic Control configuration;
- it manages the network with respect to fault detection and SLA monitoring.

The main tasks the NC has to accomplish refer to policy generation and instantiation.

4.1 Policy Generation

The NC receives a request for subscription from the RM, related to a service to be committed. The request is composed of a set of policies Network Independent (NI). The NC translates all involved policies in a network dependent format; it checks the consistency of these policies (e.g. availability of the requested resources) and sends the answer back to the RM. The generated set of policies will be stored in the NDPR.

4.2. Policy Instantiation

In this phase, the NC identifies the involved devices and sends (via COPS) the set of policies related to the request to the corresponding Device Controllers (DC). The DCs will in turn translate the received policies into the right traffic control commands needed to appropriately configure the network devices.

5 Conclusions and Future Work

In this paper we have shown an innovative approach for QoS-aware network configuration by means of policies. Such an approach has the advantage to provide a completely general way to achieve end-to-end QoS guarantees. Thanks to its layered structure, the architecture we propose is capable to make an adaptation from a service instance representation, as it is perceived at an abstract level, to a set of commands to be enforced on the underlying QoS-aware network nodes.

This architecture is going to be implemented as prototype and tested in the framework of the European project CADENUS. The main goal of this work will be:

- to emphasize the power and attractiveness of the proposed technology;
- to show its validity by means of a prototype;
- to give results of tests and trials;
- to identify current lacks and propose solutions;
- to accelerate the step toward a standardization of all the elements of policy-based management network.

References

- [CIM] Distributed Management Task Force, Inc., “*Common Information Model (CIM) schema*”, version 2.3, March 2000.
- [PCIM] J. Strassner, E. Elleesson, B. Moore and A. Westerinen, “*Policy Core Information Model - Version 1 Specification*”, RFC3060, February 2001.
- [PCIme] B. Moore, L. Rafalow, Y. Ramberg, Y. Snir, J. Strassner, A. Westerinen, R. Chadha, M. Brunner, R. Cohen, “*Policy Core Information Model Extensions*”, <draft-ietf-policy-pcim-ext-01.txt>.
- [PQIM] Y. Snir, Y. Ramberg, J. Strassner, R. Cohen, “*Policy Framework QoS Information Model*”, Internet draft, <draft-ietf-policy-qos-info-model-01.txt>, April 2000.
- [COPS] J. Boyle, R. Cohen, D. Durham, S. Herzog, R. Rajan and A. Sastry, “*The COPS (Common Open Policy Service) Protocol*”, RFC2748, January 2000.
- [SLA] S.P. Romano, M. Esposito, G. Ventre and G. Cortese, “*Service Level Agreements for Premium IP Networks*”, work in progress, Internet Draft <draft-cadenus-sla-00.txt>, available at <http://www.cadenus.org/papers>, nov 2000.
- [SLS] D. Goderis, Y. T'joens, C. Jacquenet, G. Memenios, G. Pavlou, R. Egan, D. Griffin, P. Georgatsos, L. Georgiadis, P. Van Heuven, “*Service Level Specification Semantics, Parameters and negotiation requirements*”, Internet-Draft, <draft-tequila-sls-01.txt>, work in progress, June 2001, expires December 2001.