Model-Based Management of Network Resources

Fernando Marques Figueira Filho, João Porto de Albuquerque, Paulo Lício de Geus

[1] Instituto de Computação – Universidade Estadual de Campinas (UNICAMP)
Campinas – SP – Brazil

{fernando,jporto,paulo}@las.ic.unicamp.br

Abstract. *The modelling of access control rules in terms of high-level policies has been subject of research over the last decade. Policies, in this context, define if an access is permitted or forbidden to be performed. However, they do not prescribe rules for the utilization of network resources. In this manner, a user or application might consume available resources with superfluous activities, hindering other high-priority users or applications to work properly. Following this motivation, our approach associate to a policy a set of requirements that must be fulfilled for each access. We adopt model-based management concepts, in which a policy is defined in terms of abstract entities and is represented at different levels of abstraction.*

1. Introduction

The current network environments incorporate several mechanisms in order to provide access control at network level, such as packet filters, VPNs etc. Thus, the complex configuration and maintenance of those mechanisms motivated a prolific research branch that is based upon the use of policies to model access control rules, in terms of abstract entities like users, roles or services.

One approach in this direction is the *model-based management* (MBM) introduced in [Lück et al. 2001]. It supports the construction of policy hierarchies [Sloman 1993], so that an initial set of high-level abstract policies can be refined through intermediate levels until reaching computer executable policies.

Policies, in MBM, define if an access is permitted or forbidden to be performed. However, once an access is permitted, a user can consume available network resources with superfluous activities, possibly stealing resources from high-priority users or services. In such cases, a counter-measure is to forbid these resource-consuming activities, by blocking their access. When such measure is impossible or considered too drastic, another solution is to limit the network resource usage for certain classes of users or applications.

Following this motivation, our purpose is to extend the MBM model to support policies that do not only prescribe if an access can be performed, but also *how* it should be performed. To a policy we associate a set of requirements that must be fulfilled for each access concerning that policy. These requirements could be restrictions like maximum bandwidth consumption or performance guarantees such as maximum packet delay.

To pursuit this goal, we adopted the architecture of *differentiated services* standardized by IETF, which can provide customized service characteristics for different traffic flows. Our main goal is to create an architecture that manages a differentiated service

network based on a set of high-level policies, separating the configuration details from the management needs of a system.

In the following section we give an overview of the MBM model, as well as a detailed explanation concerning our policy modelling and refinement process. Subsequently, we introduce an approach for applying policy requirements in a differentiated services enabled network domain.

2. Model-based Management

The *model-based management* was initially proposed in [Lück et al. 2001] and aims to support the construction of policy hierarchies by using an object-oriented representation of the system. The model is divided into three layers. Each layer is a representation of the managed system in a given level of abstraction.

The uppermost level is called *Roles & Objects* (RO) and offers a business-oriented view of the system. It is based on concepts from Role-Based Access Control (RBAC) [R. S. Sandhu and Youman 1996]. The main entities are: *Roles*, in which people who are working in the modelled environment act; *Objects* of the modelled environment which should be subject to access control; and *AccessModes*, representing the ways of accessing objects. The entity *AccessPermission* associates all those entities in order to compose an *authorization policy* [Sloman 1993], that allows the performer of a *Role* to access a particular *Object* in the way defined by an *AccessMode*.

The second level (SR) offers a system view defined from the standpoint of the services that the system provides. Entities of this level represent: people working in the modelled environment (*User*); subjects acting on user's behalf (*Subject*); *Resources* in the network; and *Services*, used to access *resources*.

The lowest level (PH) is responsible for modelling host processes and their communication. *ProtocolPermissions* allow the transition of packets between processes. Several other entities are also defined into the three layers, but they will not be mentioned here for the sake of brevity.

Policies are, in MBM, subject to an automatic refinement when descending the abstraction levels of the model. In this manner, each *AccessPermission* is refined into one or more *ServicePermissions* in the SR level, which expresses an authorization for a *Subject* (on behalf of a *User*) to use a *Service* to access a *Resource*. Subsequently, the *ServicePermissions* are refined into a set of *ProtocolPermissions* at the PH level.

3. Modelling and Enforcing Policy Requirements

A MBM policy defines if an access is permitted or forbidden to be performed. In order to extend the MBM model, our policy representation is defined by associating requirements to the MBM's authorization policy structure (see Sec. 2.). This requirements must be fulfilled for each access authorized by a policy.

Requirements are also subject to the refinement process. At the uppermost level (RO) they are represented by abstract service names, such as *"medium quality video-conference service"* or *"low priority file sharing service"*. Each one of these abstractions is refined into a set of SLSs (*Service-Level Specifications*) at the SR level. A SLS specifies the technical details of a SLA (*Service-Level Agreement*). A SLA is a contract established

between a service provider and a customer, defining the expectations and obligations that exist in their business relationship. Since we are concerned only about computer network services, SLS information is sufficient to describe the requirements of a policy.

Finally, the way in which a SLS is refined to the lower level (PH) will depend on the technology used to enforce our policy requirements. The next section shows an approach based on *Differentiated Services* over IP networks.

3.1. SLS Enforcement in a Differentiated Services Architecture

Differentiated Services [Blake et al. 1988] proposes a basic method to differentiate traffic among network nodes. Traffic entering a network is classified and marked by assigning a code to each IP packet. Packets marked with the same code will receive a particular per-hop forwarding behavior (PHB) on nodes along their path. A PHB defines how an individual router will treat an individual packet when sending it to the next hop through the network.

However, SLS definitions are applicable to an access performed between two nodes whose path could encompass several routers. A special treatment must be applicable in the entire path that connects source and destination nodes. Thus, we say that a SLS defines an end-to-end flow behavior.

Since a PHB is applicable only for a single router, we have to compose a set of PHBs in order to construct an end-to-end flow behavior that enforce the requirements expressed by a SLS. For this purpose, we require two types of information: topology and run-time performance measurements. Topology information represents router-to-router connectivity. A path from a source router and a destination router must be provided. Performance information can be obtained by combining performance parameters of each router in a path. The next section introduces some model support to pursuit this goal.

3.1.1. Extracting topology information

Our approach to extract the topology information uses an extension of the MBM model introduced in [Porto de Albuquerque et al. 2005], called *Diagram of Abstract Subsystems* (DAS). A DAS is a graph that represents the system architecture, establishing thereby the possible communication flows between pairs of source and destination nodes.

Authorization policies in DAS are represented by a set of *ATPathPermission* objects, which are not directly modelled by the user, but rather automatically refined from the *ServicePermissions* and their related objects of the SR level (see Sec. 2.). Each *ATPathPermission* is a path in the graph and represents the permission for an *actor* (source node, in DAS terminology) to reach a certain *target* (destination node) passing through the required *mediator* nodes (routers). In this manner, the topology information can be easily derived from each *ATPathPermission* object, by selecting all mediators along the path.

3.1.2. Performance measurements

Each router has performance parameters observed locally. The parameters include number of dropped packets, minimum and maximum rates of packet transmission etc. These parameters are calculated and maintained for each PHB it processes. Our purpose is to

develop a management architecture that collects this performance parameters at run-time for each router in a given path and produces adequate configurations in order to enforce SLS parameters and fulfill policy requirements.

To illustrate our idea, consider a SLS that specifies a maximum drop rate of 30% for a given access. Suppose we found three mediators along the path obtained from the corresponding *ATPathPermission*. We have to configure each router in a coordinated manner, such that the overall drop rate for that single flow A do not go over 30%. This is not a trivial calculation, since each router maintains performance measurements for each PHB, not for each source/destination pair. For each router, several other flows could be classified with the same PHB, so suppose we have another flow B (with different source/destination pair) classified with A's PHB and whose path encompasses only the same three routers. If B's SLS specifies a drop rate of 20%, then we have to generate a configuration that guarantees a maximum accumulated drop rate of 20% among those routers. Since flows A and B are mixed together in the same PHB classification, we have always to consider the lowest rate per PHB.

4. Future Work

Our future efforts will be centered around the development of a management architecture capable of enforcing policies and their requirements in a coordinated manner. Management tasks will use low-level information produced by MBM's refinement process (see Sec. 2.), such as PH level policies and DAS topology information. A central manager will collect performance measurements in order to produce new configurations and assert fulfilment of SLS requirements. A protocol such as SNMP should be used for distributing and collecting management information.

Since we assume the sharing of network level resources among several different flows, a special case that must be considered is when there is insufficient resources for the enforcement of a given set of policies. Algorithms like *max-flow-min-cut*, when applied in a DAS graph, could detect bottleneck routers and possibly point some optimizations such as rerouting of traffic flows. At a higher abstraction level, policy requirements could be negotiated by offering a similar service or demoting the quality of service dedicated to low priority users or services in order to transfer network resources to higher priority ones.

References

Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and Weiss, W. (1988). An architecture for differentiated service. RFC 2475, IETF.

Lück, I., Schäfer, C., and Krumm, H. (2001). Model-based tool-assistance for packet-filter design. In *POLICY '01: Proceedings of the International Workshop on Policies for Distributed Systems and Networks*, pages 120–136, London, UK. Springer-Verlag.

Porto de Albuquerque, J., Krumm, H., and de Geus, P. L. (2005). Policy modeling and refinement for network security systems. In *IEEE 6th International Workshop on Policies for Distributed Systems and Networks*, Stockholm, Sweden.

R. S. Sandhu, E. J. Coyne, H. L. F. and Youman, C. E. (1996). Role-based access control models. In *IEEE Computer*.

Sloman, M. (1993). Policy hierarchies for distributed systems management. In *IEEE Journal on Selected Areas in Communications*, pages 1404–1414.