DC

Interlibrary Loan/Document Delivery
Picklist Report (Lending)

Lending No.: 8581400 **Printed Date:** 19-APR-2016 **Status:** In Process/En traitement

Expiration Date: 25-APR-2016

Title: Internet quality and performance and control of network systems : 6-7 November 2000, Bo USA ; SPIE proceedings series ;; v. 4211;

Format: Conference Proceedings/Compte-rendu de conférence

Author:

Edition:

Publisher Info: ;

Publication Date: 2001

ISBN/ISSN: 0819438766 / 0277-786X

Media Type: Imprimé/Printed

Article Title: Managing MPLS/VPNs with policies

Article Author: Omar Cherkaoui ; Mounir Boukadoum and Alain Sarazin

Volume/Issue: 4211 **Pages:** ?? **Article Date:** 2001

Copyright Compliance: Cette copie est requise par un usager autorisé pour une utilisation équitable

Request Notes:

Need By:

Requesting Library: UQAM-Bibliotheque des sciences **Supplying Library:** University of Waterloo, Library

Requester ID: QMUQS

NLC-BNC Code: NLC-BNC:QMUQS

Ariel Address: 132.208.68.254

Email: colombo.peb.sciences@uqam.ca

Requester ILL #: 1761165

Patron Name:

Barcode:

Patron Category:

Patron Department:

Service Type: Copy/Copie

Delivery Method: Article Exchange

Pickup Location:

Managing MPLS/VPNs with policies

Omar Cherkaoui*, Mounir Boukadoum, and Alain Sarazin
Université du Québec à Montréal, Montréal, Canada
cherkaoui.omar@uqam.ca

Abstract

This paper proposes an approach for the dynamic management of MPLS-based VPNs. MPLS and VPNs significantly contribute to achieve QoS within networks but there remain dynamic management problems associated with their use. We believe that these problems can be solved by using a policy model; such an approach also enables subscribers to keep control of their VPNs and share information with service providers. We used a PCIM-enabled network model to account for the peculiarities of the two technologies and combined the resulting schema with COPS and the necessary policy tools. The resulting framework was then tested on a MPLS network. The results show that, with some limitations, the approach does provide the expected functionality.

Keywords: VPN, MPLS, PCIM, CIM, DEN, COPS, PBN, Policy Management.

1. Introduction

Over the past years, QoS (Quality of Service) considerations have become key issues in network applications. New applications such as Voice over IP (VoIP), video streaming, conference calling, and other multimedia applications heavily depend on the QoS provided by the underlying network. The increased convergence of voice and data communication networks will most certainly require an upgrade of the legacy IP technology in order to support new QoS and service differentiation mechanisms.

To address these issues, great expectations have emerged from the potential of associating the MPLS (Multiple Protocol Label Switching) [1] protocol architecture with VPNs (Virtual Private Networks) [2]. While each of these technologies is promising on its own, their combination holds even greater promises for end customers and telecommunication service providers, as it will provide cost-effective and high quality communication capabilities. Unlike regular VPNs, those based on MPLS may be deployed over an IP network, gaining all the advantages offered by the TCP/IP protocols while retaining all those offered by frame relay, ATM and such present-day supports. Therefore, the interesting question becomes "how can we extend these benefits over a link spanning multiple VPNs"? Using policy servers with the PCIM model may offer an easy solution to the problem of managing such multiple VPN networks. This Policy-Based Networking (PBN) [3] signals a shift in the way networks are controlled and managed.

Considering all the various mechanisms and protocols that are needed to provide VPN-based QoS in IP networks, the task of managing and coordinating them across a network may constitute a formidable challenge if done manually. It would be difficult, if not impossible, to configure every network device with the right queuing and traffic processing mechanisms to

* Correspondance : email : Cherkaoui.omar@uqam.ca; http://www.info.uqam.ca/~cherkaou/; Telephone : 514-987-3000 ext 3513; Fax : 514-987-8477

Internet Quality and Performance and Control of Network Systems,
Angela L. Chiu, Frank Huebner-Szabo de Bucs, Robert D. van der Mei, Editors,

provide consistent, priority-based service everywhere in a large scale network. This may be true even if the entire network consisted of equipment from just one vendor, and most large networks are made up of heterogeneous equipment from multiple vendors, each supporting different mechanisms and its own configuration methods; so, the problem becomes even more complex. In addition, QoS applications must continue to work properly in the face of dynamic network and organizational changes – situations that existing, "traditional" network management applications are not well suited to handle.

In addition to the abundance of information, a QoS configuration is also rather dynamic. Applications that need QoS capabilities from the network come and go – such as VoIP calls, video conferences, etc. – and requests for network resources from applications do not specify an a priori destination. Hence, different locations in the network require different QoS mechanisms to be implemented, at different times.

To alleviate these problems, we have designed a management model that makes use of a policy server based on the PCIM model now managed by the DMTF (Distributed Management Task Force) group [4]. This server, along with its graphical editor, allows us to manage the various aspects of service policies over a network. In particular, it can be configured to set up MPLS tunnelling and, therefore, VPNs.

The remainder of this paper is organized as follows. Section 2 outlines the technologies used: VPNs, MPLS, PCIM and the Policy Server. Section 3 then explains how these can be put to use in setting up a MPLS VPN that can span multiple VPNs. Finally, section 3 provides a example of use of the proposed management model. Finally, section 4 summarizes the experience we gained during the development and design phases of this work.

2. MPLS/VPN Architecture

In this section, we present the two underlying technologies VPN and MPLS.

2.1 Virtual Private Networks

Consider a set of "sites" which are attached to a common network that we may call the "backbone". Let us apply some policy to create a number of subsets of that set, and let us impose the following rule: two sites may have IP interconnectivity over that backbone only if at least one of these subsets contains them both. The subsets we have created are "Virtual Private Networks" (VPNs). Two sites have IP connectivity over the common backbone only if there is some VPN that contains them both. If all the sites in a VPN are owned by the same enterprise, the VPN is a corporate "intranet". If the various sites in a VPN are owned by different enterprises, the VPN is an "extranet". A site can be in more than one VPN; e.g., in an intranet and in several extranets. In general, when we use the term VPN we will not be distinguishing between intranets and extranets.

If the backbone is owned and operated by one or more Service Providers (SPs), the owners of the sites are the "customers" of the SPs. The policies that determine whether a particular collection of sites is a VPN are the policies of the customers. Some customers will want the implementation of these policies to be entirely the responsibility of the SP. Other customers may want to implement these policies themselves, or to share with the SP the responsibility for implementing these policies. In this document, we are interested in mechanisms that may be used to implement these policies.

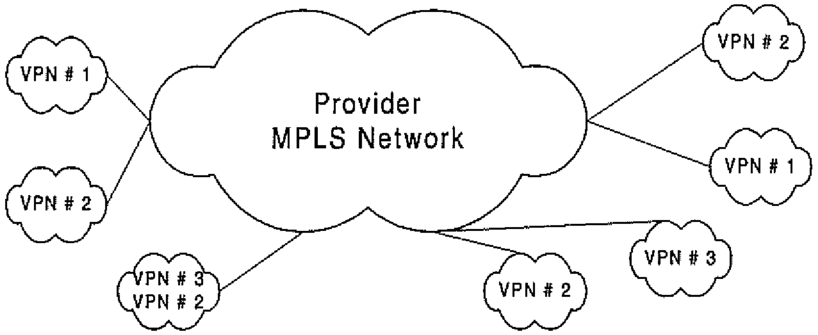


Figure 1

The model we use consists of a provider network backbone with customer sites around the edges. Each customer site is associated with one or more VPNs. Figure 1 shows an example configuration with seven customer sites and three VPN.

Thus, VPNs are a way to set up the equivalent of a private network over a public one, gaining advantages relative to cost and reach. They provide WAN communications more cheaply and more globally, which is to say they offer the best of both worlds: the security, bandwidth and quality of service (QoS) guarantees typically associated with private networks and the flexibility, pervasiveness and low cost of TCP/IP networks. Still, they raise several challenges. Among these is how to deal with issues of QoS.

2.2 Multiprotocol Label Switching

As most of the current efforts to insure the convergence of voice, data, and multimedia networks use IP-based protocols, a need exists for technical and operational improvements. Improving the original TCP/IP architecture, whether to differentiate vendor products, or to create integrated public networks, has become a significant industry incentive. Efficiency enhancements that improve switching price/performance and lower overall costs (which could stimulate the use of voice over IP, for example) are eagerly anticipated. Multi Protocol Label switching is one of the industry's responses to these challenges. This IETF standard has become a key technology to the future of large-scale IP networks. MPLS allows the development of IP networks that are QoS enabled. Packets that enter a MPLS cloud are labelled by edge routers to identify their path. Routers within the cloud then use the labels to switch the packets through a label-switching path (LSP).

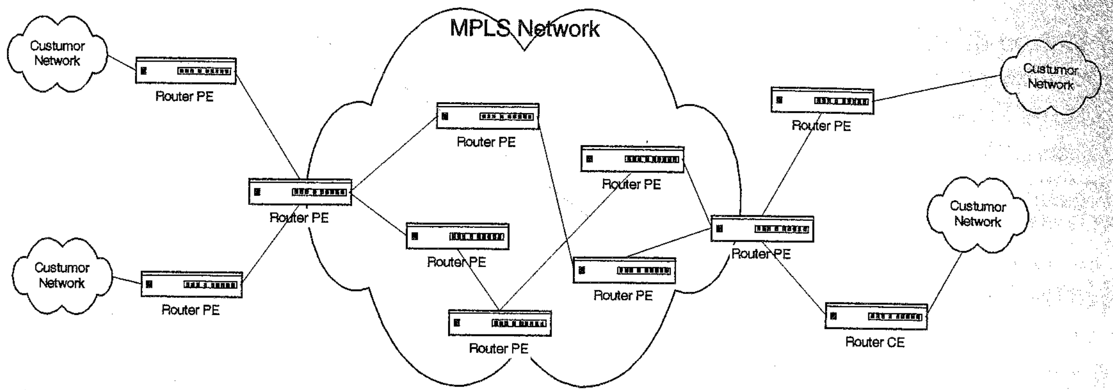


Figure 2

Figure 2 provides the example of an MPLS network where customers communicate via an MPLS cloud that connects provider edge (CE) routers to customer edge (PE) routers.

3. Policy framework

This section provides an overview of the services and protocols defined by IETF, and identifies the key entities involved. It provides a context for understanding the structure of the Policy Management.

3.1 Policy Based Networking

PBN [5] enables the coordination of network information, and dynamically maps it to configuration information, including queuing mechanisms, packet treatment methods, link capacity based on service class, etc. PBN mechanisms such as policy servers can automatically identify the various devices in the network, and determine which QoS capabilities they support. Protocols such as Common Operation Policy Services (COPS) [6] are used to send the appropriate configuration information to the devices, and to allow the network devices to efficiently provide feedback about the state of the network to the PBN system. This feedback is an essential component of PBN for dealing with the dynamic nature of network operations.

In the COPS scenario, the PE router acts as a policy enforcer that sends requests to a policy decision point, the policy server. The latter processes each request according to the stored policy in its LDAP directory and returns a decision to the PE on whether the policy is to be enforced. As all PE routers do not necessarily support COPS, a second scenario involves a daemon that acts as a policy enforcer, converting the policy in CLI to activate the action on the PE.

Figure 3 illustrates the mechanism of the policy enforcement. It shows a policy editor that stores policies in an LDAP server [7] to be accessed by both a COPS server and a policy enforcer. In the COPS case, requests from either of the two PE routers are processed by the COPS server and decisions are returned; in the second case, the policy enforcer directly sends CLI commands to the PE routers.

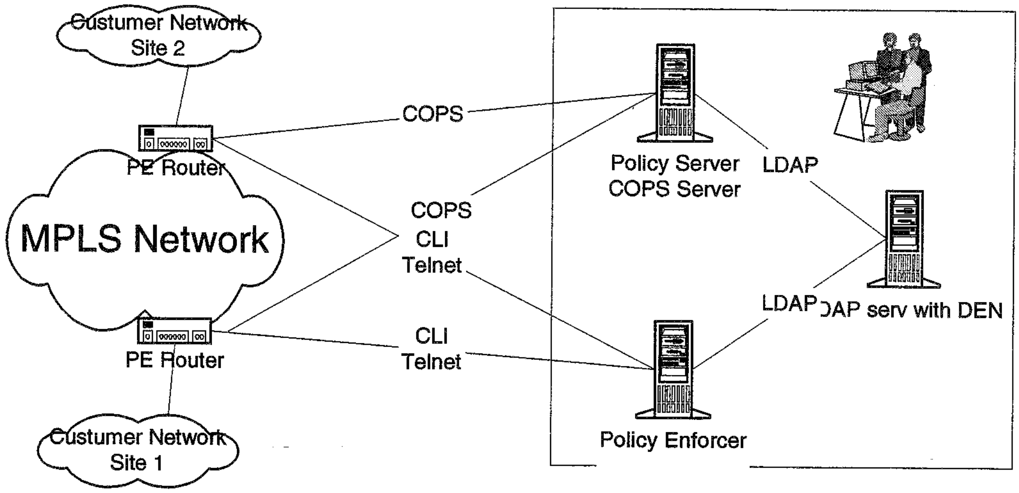


Figure 3

3.2 Policy Core Information Model

Policy Core Information Model (PCIM) [8] is an object-oriented information model for representing policy information currently under joint development in the IETF Policy Framework WG and as extensions to the Common Information Model (CIM) activity in the Distributed Management Task Force (DMTF). This model defines two hierarchies of object classes: structural classes representing policy information and control of policies, and association classes that indicate how instances of the former will define mappings of this information model to various concrete implementations, for example, to a directory that uses LDAPv3 as its access protocol.

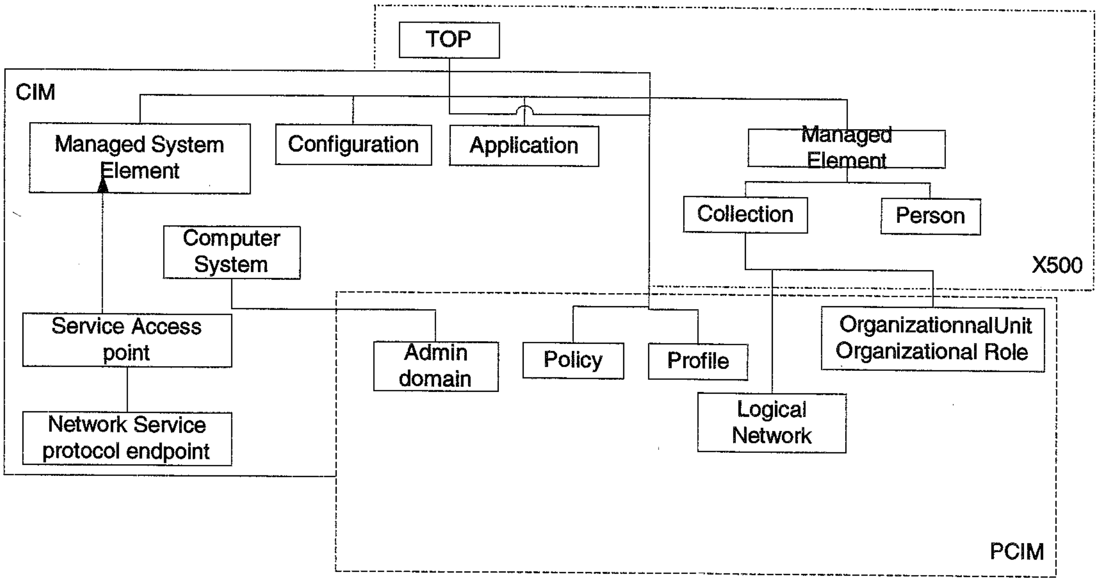


Figure 4

PCIM allows us to represent users, equipments and services alike as objects that interact according to predefined policies. Users and applications are associated with profiles, that is, with a set of attributes and behaviours that indicate what an object does without specifying how. The model is composed of three parts: six base classes (Network Device, Network Protocol, Network Media, Profile, Policy and Network Services), object-oriented mechanisms to ensure extendibility, and relations between objects.

The policy classes and associations defined in this model are sufficiently generic to allow them to represent policies related to anything. However, it is expected that their initial application in the IETF will be for representing policies related to QoS, IPSec, and VPN. Policy models for application-specific areas such as these may extend the Core Model in several ways.

The following describes the classes that we used in our application.

4 Modelling VPN networks for MPLS

In order to develop our management model, we focused on insuring agreement between the semantics and the enforcement of an MPLS policy, and on allowing the derivation of task-specific representations that will be used to configure MPLS-enabled edges. To achieve these goals, a model of an MPLS policy was devised.

4.1 Information model

Figure 5 provides the architecture of our management model. It shows a set of administrator-configurable information objects using UML notation. The model relies on deriving a new class, called MPLS-VPN, from PCIM's Top class and on creating a subclass of PCIM's PolicyAction class. Itself derived form Top. The management information model is mainly specified by classes derived from MPLS-VPN; these provide information on customers, providers, as well as on MPLS routing policies. In addition, class MPLS_VPNAction identifies actions to be performed when the conditions of a given policy are met.

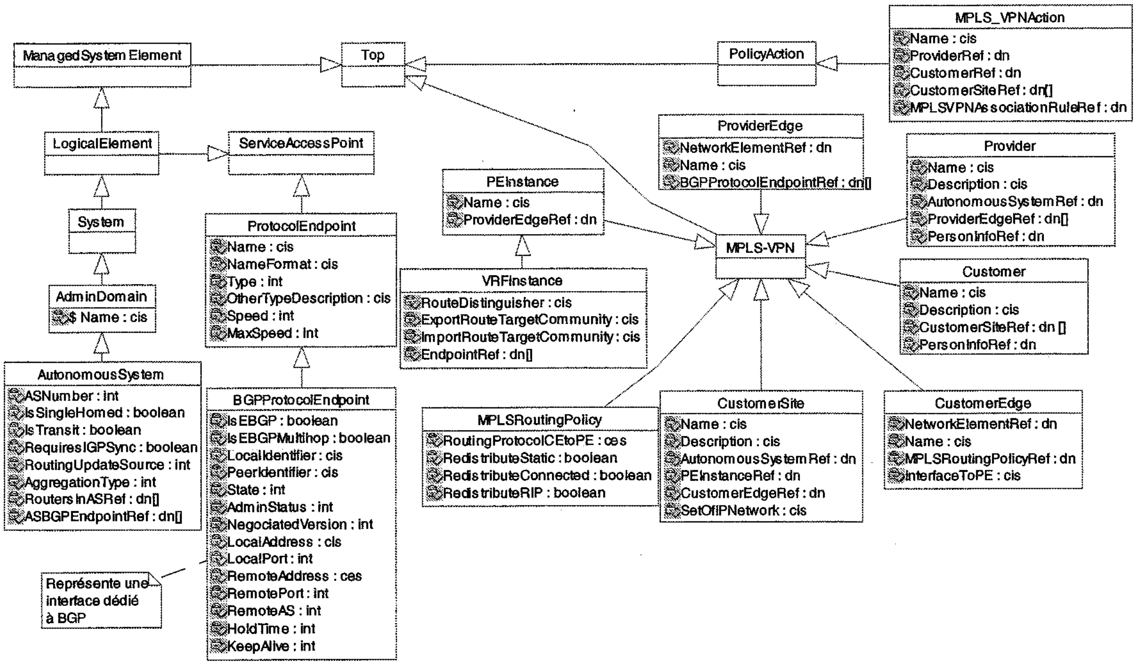
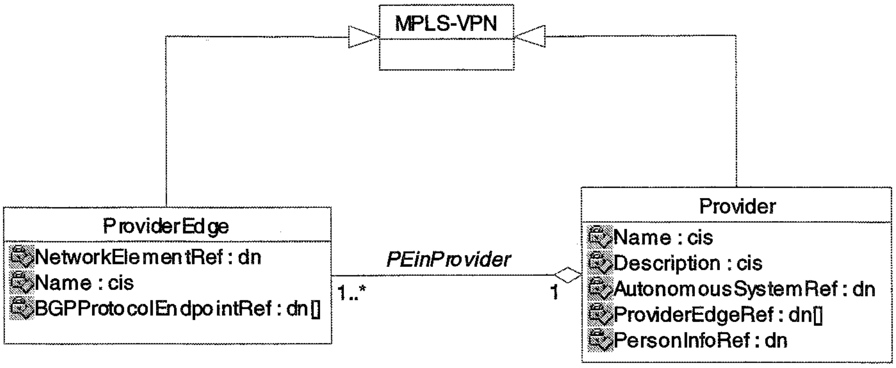


Figure 5. Overall architecture of the MPLS-VPN management mode

4.2 MPLS-VPN classes

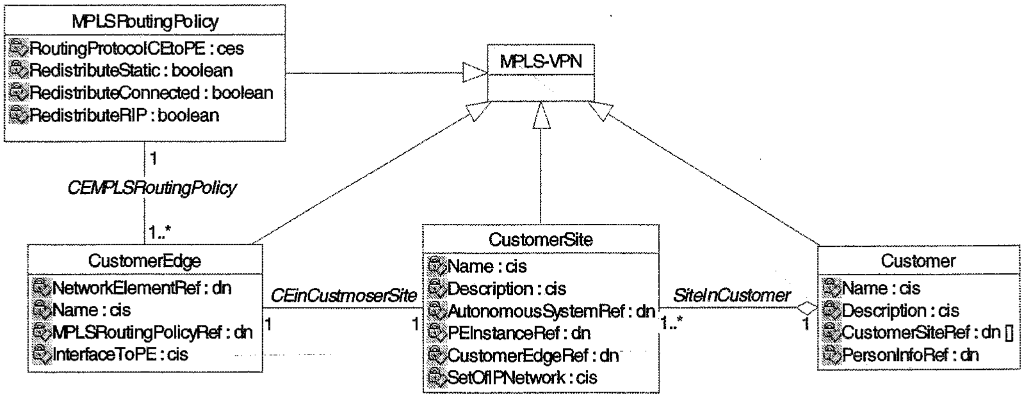
4.2.1 Provider-related classes

There are two such classes, the first one represents the MPLS-VPN service provider and the second one represents the MPLS edge routers that the provider connects to, the VPN I/O points in other words. A provider can have several edge routers each of which being uniquely associated with it.



4.2.2 Customer-related classes

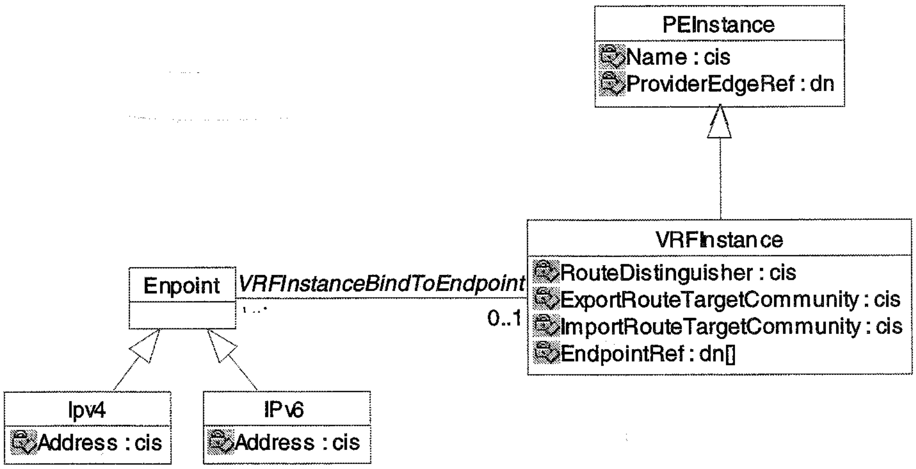
There are four such classes. Two of them serve a similar purpose to the provider-related classes: identify the customer and its associated edge router. In addition, a third class describes the customer site in terms of IP addresses, its edge router and the provider edge router that its router connects to, and a fourth class describes the routing policy that exists between the customer and provider edge routers. A customer may have several sites and a routing policy may apply to several customer edge routers. On the other hand, there is only one edge router per customer site, and one routing policy per edge router.



A MPLS RoutingPolicy object specifies the routing protocol between CE and PE routers (BGP, Static or RIP), and whether the provider edge should redistribute routing information to the customer edge. Reference to the provider edge within a customer site object is indirect via an object of the PEInstance class. This class frees the customer from knowing the actual location of a provider edge.

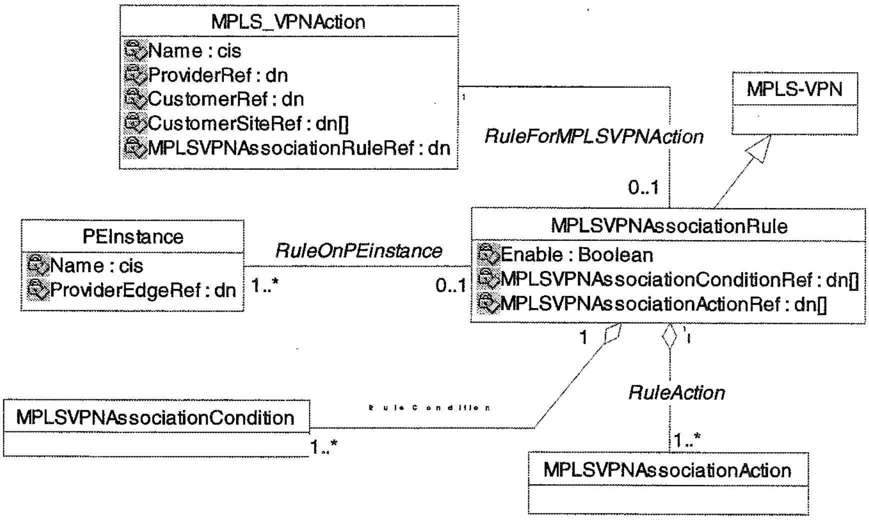
4.2.3 Interface classes

PEInstance is also the root class for describing a VPN routing mechanism as describes by instances of the VRFInstance class. A VRFInstance object implements a virtual router within a PE router. Its purpose is, using BGP, to identify import and export routes to be shared by PE routers within the same VPN. The RouteDistinguisher attribute allows the creation of distinct routes for common IPv4 addresses; when concatenated with an Ipv4 address, it gives birth to a new address class, VPN-Ipv4.



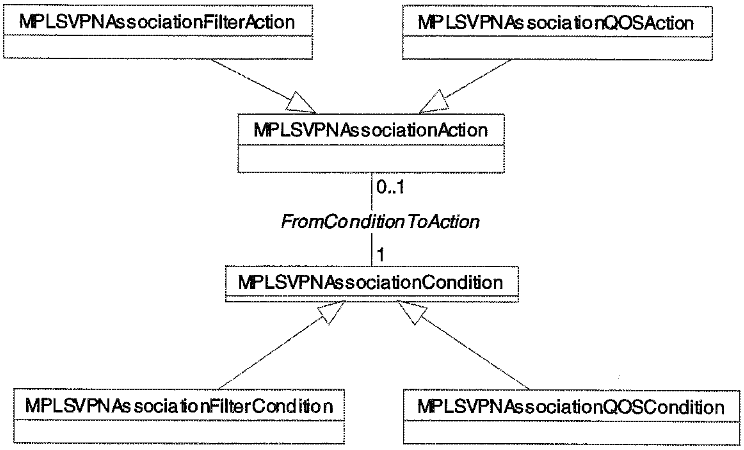
5. MPLS-VPN service classes

These classes all derive from the MPLS_VPNAction class, itself derived from the DEN PolicyAction class (see figure 5). MPLS_VPNAction is the core class for the MPLS_VPN services that the provider deploys on behalf of the client, in addition to identifying the two parties and the customer sites that are part of the VPN (at least two), it provides a reference to the rules used to start actions when policy conditions are met. There is one MPLS_VPNAction object for each VPN that a provider and customer associate with.



The rules consist of associations between conditions and actions to be performed; they are stored in MPLSVPNAssociationRule objects as two sets of references, one to MPLSVPNAssociationCondition objects and the other to corresponding MPLSVPNAssociationAction objects. An MPLSVPNAssociationRule object is only active if its Enable data member has a value of TRUE; it is disabled otherwise. An MPLS-VPN network may have no rule or one or more rules that apply to one or several PE instances. Also, a rule may contain one or more conditions, each of which related to one or more actions. Finally, an action may be associated with more than on rule.

5.1 Classes related to the conditions and actions of MPLS-VPN services


MPLSVPNAssociationFilterAction
MPLSVPNAssociationQOSAction
MPLSVPNAssociationAction
0..1
FromCondition ToAction
1
MPLSVPNAssociationCondition
MPLSVPNAssociationFilterCondition
MPLSVPNAssociationQOSCondition

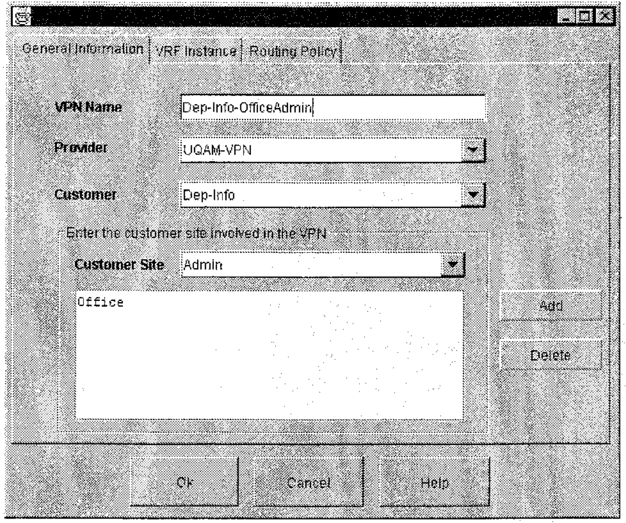
These MPLSVPNAssociationCondition and MPLSVPNAssociationAction classes specify the criteria to apply for validating a given condition and the actions to take upon meeting a given condition. They supply information for associated filter and QoS subclasses. The structure of these classes is still under development.

6. Implementation of the model

For the realization of the VPN-MPLS management architecture, we used a similar platform to that of figure 3. The development of the different components was accomplished using the java programming language. The system includes a COPS policy server and a policy enforcer. Policies are entered via an editor in the network administrator's console and they are saved in a LDAP server. As mentioned before, two mechanisms of policy enforcement are possible: COPS and a text-based protocol (CLI commands).

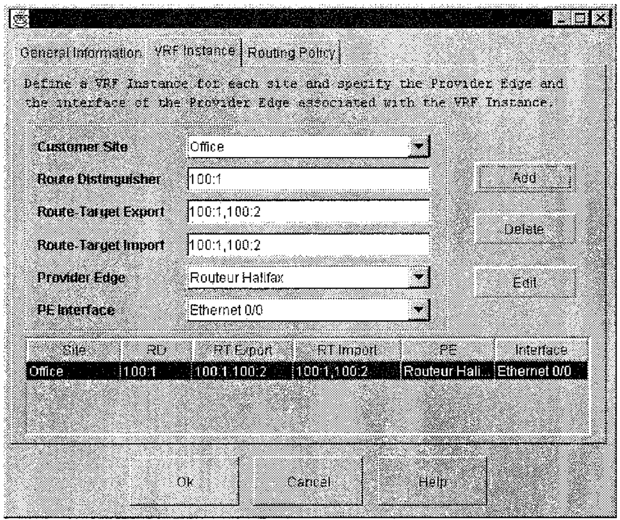
6.1 Example of used interface for the policy editor

The following example provides screen snapshots of a configuration where an MPLS-VPN network, a customer site and a CE are defined. Customers, Providers and PEs are defined in a similar fashion. Also included are snapshots of the policy editor main window and of defining a scheduler and a schedule



VPN Name: Dep-Info-OfficeAdmin
Provider: UQAM-VPN
Customer: Dep-Info
Enter the customer site involved in the VPN
Customer Site: Admin
Office

MPLS-VPN creation

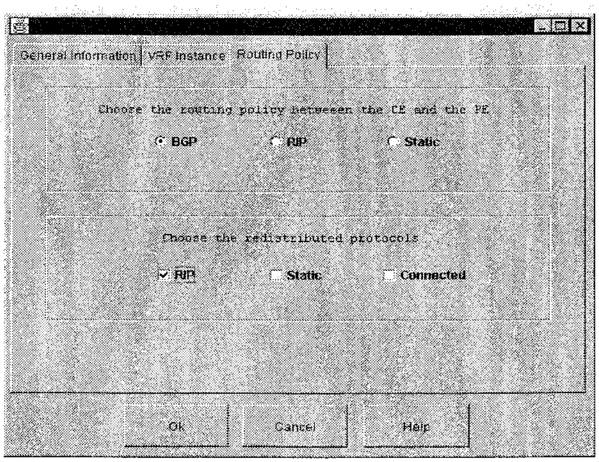


Define a VRF Instance for each site and specify the Provider Edge and the interface of the Provider Edge associated with the VRF Instance.

Customer Site: Office
Route Distinguisher: 100:1
Route-Target Export: 100:1,100:2
Route-Target Import: 100:1,100:2
Provider Edge: Routeur Halifax
PE Interface: Ethernet 0/0

Site	RD	RT Export	RT Import	PE	Interface
Office	100:1	100:1,100:2	100:1,100:2	Routeur Hali...	Ethernet 0/0

Associations between VRFs and VPN sites



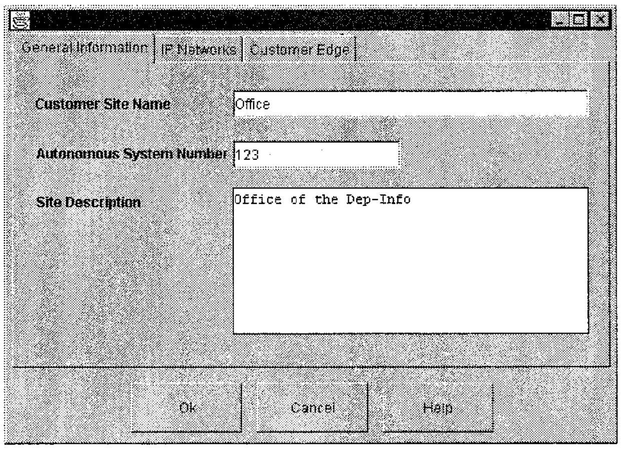
Choose the routing policy between the CE and the PE

● BGP ○ RIP ○ Static

Choose the redistributed protocols

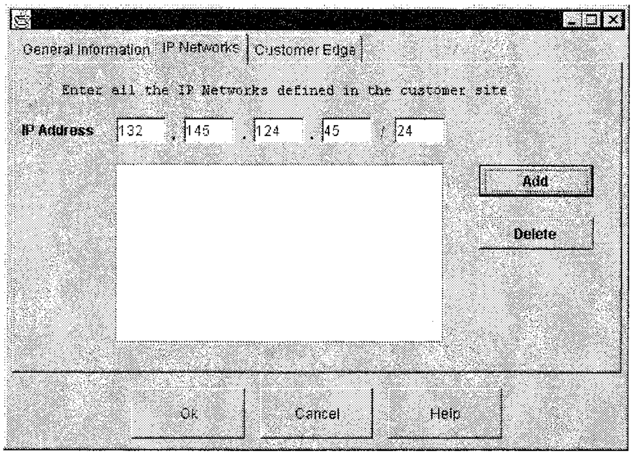
☑ RIP □ Static □ Connected

Routing policy specification between PE and CE



Customer Site Name: Office
Autonomous System Number: 123
Site Description: Office of the Dep-Info

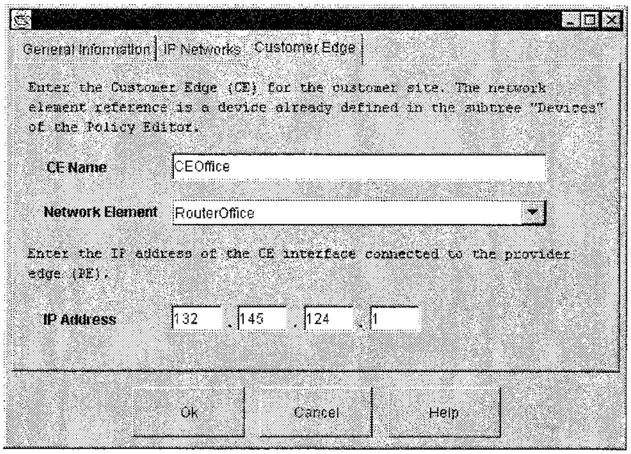
Customer site definition



Enter all the IP Networks defined in the customer site

IP Address 132 . 145 . 124 . 45 / 24

IP addresses in customer site



Enter the Customer Edge (CE) for the customer site. The network element reference is a device already defined in the subtree "Devices" of the Policy Editor.

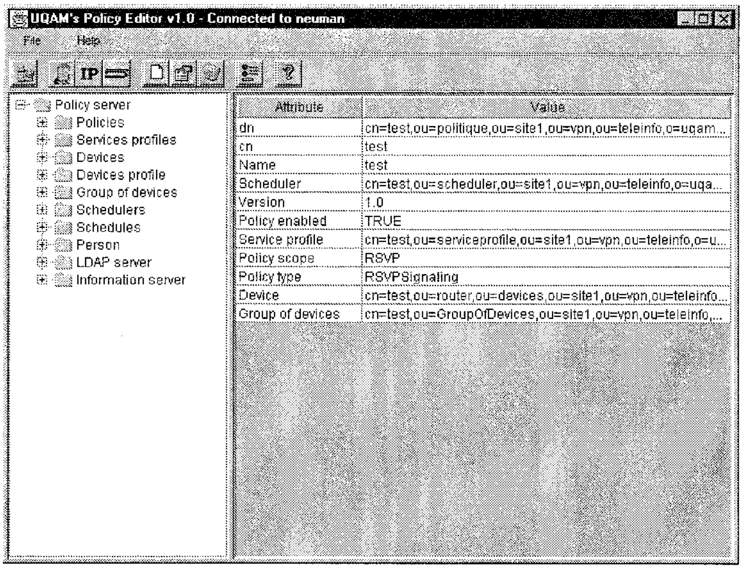
CE Name: CEOffice

Network Element: RouterOffice

Enter the IP address of the CE interface connected to the provider edge (PE).

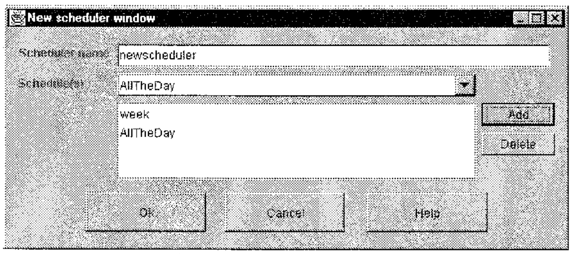
IP Address 132 . 145 . 124 . 1

CE specification



Attribute	Value
dn	cn=test,ou=politique,ou=site1,ou=vpn,ou=teleinfo,o=uqam...
cn	test
Name	test
Scheduler	cn=test,ou=scheduler,ou=site1,ou=vpn,ou=teleinfo,o=uqa...
Version	1.0
Policy enabled	TRUE
Service profile	cn=test,ou=serviceprofile,ou=site1,ou=vpn,ou=teleinfo,o=u...
Policy scope	RSVP
Policy type	RSVPSignaling
Device	cn=test,ou=router,ou=devices,ou=site1,ou=vpn,ou=teleinfo...
Group of devices	cn=test,ou=GroupOfDevices,ou=site1,ou=vpn,ou=teleinfo,...

Policy editor main window

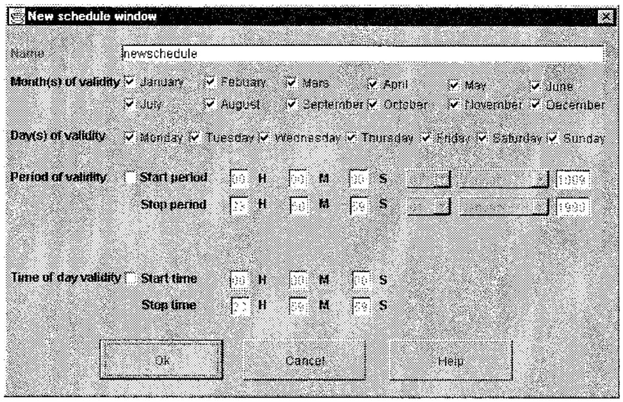


Scheduler name: newscheduler

Schedule(s): AllTheDay

week
AllTheDay

Example of defining a scheduler



Name: newschedule

Month(s) of validity ☑ January ☑ Febuary ☑ Mars ☑ April ☑ May ☑ June ☑ July ☑ August ☑ September ☑ October ☑ November ☑ December

Day(s) of validity ☑ Monday ☑ Tuesday ☑ Wednesday ☑ Thursday ☑ Friday ☑ Saturday ☑ Sunday

Period of validity ☐ Start period 00 H 00 M 00 S 1999
Stop period 23 H 59 M 59 S 1999

Time of day validity ☐ Start time 00 H 00 M 00 S
Stop time 23 H 59 M 59 S

Example of defining a schedule

7. Results and discussion

Testing of our model was undertaken using Cisco's 2611 routers and 5509 catalyst switches, running under the 12.1 enterprise IOS. These equipments support COPS but don't offer the capability to directly implement the objects in our information model. Consequently, they were programmed using Cisco's CLI.

Our experiments helped us improve the completeness of the objects that can be used to configure routers for VPN establishment, in distinction with simply using CLI. When comparing the two approaches, it appears that using CLI provides for easier results. On the other hand, CLI is version dependent and, therefore, is not evolutive.

Our experimentations helped us develop our information model, as well as validate different approaches to enforce policies. We defined efficient object classes to achieve network configuration. On the minus side, it is not yet clear to us how to use them to perform efficient real-time traffic engineering [9, 10]. Upon completion of our work on the MIB, we expect to be in a better position to undertake the validation of our information model to classify MPLS packets [11].

8. Summary

We described an approach for the dynamic management of MPLS-based VPNs. Using a policy model helps us solve the problem while enabling subscribers to keep control of their VPNs and share information with service providers. We used a PCIM-enabled network model and combined the resulting schema with COPS and the necessary policy tools. The resulting framework was tested on a MPLS network and provided promising results.

9. References

[1] Rosen, E., Viswanathan, A., Callon, R., "Multiprotocol Label Switching Architecture", work-in-progress, draft-ietf-mpls-arch- 06.txt, August 1999.

[2] Malis, A., Muthukrishnan, K., "Core MPLS IP VPN Architecture", draft-muthukrishnan-mpls-corevpn-arch-03.txt, June 2000 .

[3] Westerinen, A. Schnizlein, J., Strassner, J., Scherling, M., Quinn, B., Perry, J., Herzog, S., Huynh, A., Mark, " Policy Terminology", draft-ietf-policy-terminology-00.txt, July 2000.

[4] Distributed Management Task Force, www.dmtf.org.

[5] Sloman, M., Lupu, E. "Policy Specification for Programmable Networks" Extended version of paper in Proceedings of First International Working Conference on Active Networks (IWAN'99), Berlin, June 1999, S. Covaci ed., Springer Verlag publisher.

[6] Boyle, J., Cohen, R., Durham, D., Herzog, S., Raja, R., Sastry, A., "The COPS (Common Open Policy Service) Protocol", IETF <draft-ietf-rap-cops-07.txt>, August 1999.

[7] Wahl, M., Howes, T., Kille, S., "Lightweight Directory Access Protocol (v3)", IETF RFC 2251, Proposed Standard, December 1997.

[8] B. Moore, E. Ellesson, J. Strassner, A. Westerinen, "Policy Core Information Model -- Version 1 Specification", <draft-ietf-policy-core-info-model-07.txt>, July, 2000.

[9] Awduche, D., Chiu, A., Elwalid, A., Widjaja, I., Xiao, X., "A Framework for Internet Traffic Engineering", work-in-progress, draft-ietf-tewg-framewrk-01.txt, May 2000.

[10]Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., McManus, J.,"Requirements for Traffic Engineering over MPLS", RFC 2702, September 1999.

[14]Nadeau,T., Srinivasan, C., Viswanathan,A., "Multiprotocol label Switching Packet Classification Management Information Base Using SMIv2", work-in-progress, draft-nadeau-mpls-packet-classifier- mib-00.txt, March 2000.