Using policy-based MPLS management architecture to Improve QoS on IP Network

Ruey-Shun Chen*, C.C.Chen** and H.Y. Chen**

*Department of Information Management, China technologyUniversity,
**Institute of Information Management, National Chiao Tung University,
1001 Ta Hsueh Road, Hsin-Chu city, Taiwan

Abstract

Multi-Protocol Label Switching (MPLS) is in the process of standardization by the Internet Engineering Task Force (IETF). It is regarded as a technology for traffic engineering and QoS in IP-networks.

we proposed a IETF Policy-based Network Management Framework and policies with MPLS specific classes. It uses a three-level policy architecture, which includes managing on device, network, and service level using policies for supporting Inter-serv and Diff-serve based end-to-end QoS in the Internet.

A prototyping of policy-based management system for MPLS Traffic Engineering is operating on MPLS network elements. Several experiments illustrate the efficiency and feasibility in this architecture. The results show it can reduce the time of the setup of MPLS traffic engineering tunnel over hops and MPLS traffic engineering tunnel deletion. The proposed integrated policy based management architecture will allow network service providers to offer both quantitative and qualitative services while optimizing the use of underlying network resources.

Keywords: Multiple Protocol Label Switching (MPLS), Traffic Engineering (TE), Quality of service (Qos), Policy-Based Management (PBM).

1. Introduction

The Internet Engineering Task Force (IETF) has proposed a number of QoS models and supporting technologies, including the integrated services (IntServ) and differentiated services (DiffServ) frameworks [1]. The latter has been conceived to provide QoS in a scalable fashion. Instead of maintaining per-flow soft state at each router, packets are classified, marked, and policed at the edge of a DiffServ domain. In order to achieve QoS guarantees, control plane mechanisms have been used to reserve resources on demand, but management plane mechanisms are also necessary to plan and provision the network, and manage requirements for service subscription according to available resources [2]. QoS frameworks such as IntServ and DiffServ have so far concentrated in control plane mechanisms for providing QoS. However, it would not seem possible to provide QoS without the network and service management support, which is an integral part of QoS-based telecommunications networks. Considering in particular the DiffServ architecture, a key issue is end-to-end QoS delivery. The Diff-Serv architecture suggests only mechanisms for relative packet forwarding treatment to aggregate flows, traffic management, and conditioning; by no means does it suggest any architecture for end-to-end QoS delivery. In order to provide end-to-end quantitative QoS guarantees, DiffServ mechanisms should be augmented with intelligent traffic engineering functions.
Multi-Protocol Label Switching (MPLS) is a new technology to be standardized by the IETF. The technology enables the setup of Label Switched Paths (LSPs) through an IP network. Initially, the idea of IP label switching was to speed up the packet forwarding in routers via simple table lookups instead of longest-matching prefix algorithms [3].

In this paper, we propose enhancing the IETF Policy Framework in two directions. First, we incorporate the management of Multi-Protocol Label Switching (MPLS) networks into the framework. MPLS is currently seen as a technology to influence the routing of IP networks in order to engineer the traffic with appropriate tools. QoS services are more easily and more flexible deployed in an IP-based network, because MPLS allows a network manager to pin down a route for an aggregate of flows. However, MPLS per se does not have QoS features nor mechanisms, but MPLS together with Differentiated Services (DiffServ) is the favored approach by the IETF [6] for providing IP QoS. The second enhancement of the policy framework is dealing with network-level and service-level management in IP networks.

Using MPLS networks, the notion of a Label Switched Path (LSP) brings network-level concepts into the framework which has not been dealt with in the device-level policy framework. Furthermore, using traffic engineered network, new kinds of IP services are possible. E.g., a service guaranteeing low packet loss probability can be offered using MPLS as mechanism to traffic engineer the IP network in a way that traffic is routed around hot spots. However, quantifying the service quality is very difficult. Additionally, having mechanisms for guaranteed services in place, advanced IP services need to be specified, configured, and controlled.

2. MPLS technology overview

2.1 MPLS Label Stacking

When a label is added to a packet this means that at minimum a 4 byte "shim" has been added to the packet. This shim is added between the layer 3 header and layer 2 headers. Therefore an IP packet on Ethernet would add the shim before the IP header but after the Ethernet header. MPLS forwarding is currently defined for the following layer 2 implementation: Ethernet, packet over SONET, ATM, frame-relay. MPLS has also been defined for any medium that PPP runs on top of. On most of these layer 2 implementation a label consists of a 20 bit number. The shim that is added to the packet contains more then just a label. Here is a diagram of a MPLS shim [13].

2.2 MPLS Diff-Serv-aware Traffic Engineering

The concepts of applying policy-based management mechanism are chosen to manage an MPLS network, because this is an appropriate way of dealing with large sets of managed elements. Using policy-based management for networks and systems has become very popular since the early work on policies such as [2, 3, 4]. Nowadays, some commercial products are available, which use some form of policies to configure and control networks. In the IETF there is a Policy Framework Working Group [5], which aims at resolving issues related to policy-driven management of IP networks. It includes the definition of a policy framework and information models for DiffServ, IntServ, and IP Devices.

3. IETF policy framework

The IETF Policy Framework is under development by the IETF Policy Framework working group. The framework consists of Policy Enforcement Points (PEP), Policy Decision Points (PDP), management console, and a directory to store policies together with user/network resource information as Figure2. PEPs are basically network elements and the PDP is typically referred to as the Policy Server

(PS). The components are linked by the following protocols and languages.

The Common Open Policy Service (COPS) protocol [7] is used to forward requests from PEPs to the central policy server and to pass back corresponding policy decisions. and support for reliability using TCP and keep-alive messages. Note that just recently an initiative to use policies in the SNMP framework has been started in the Configuration Management with SNMP (SNMPconf) working group [9]. A Policy Definition Language (PDL) is used to define new policies in terms of policy rules with condition and action lists [15].

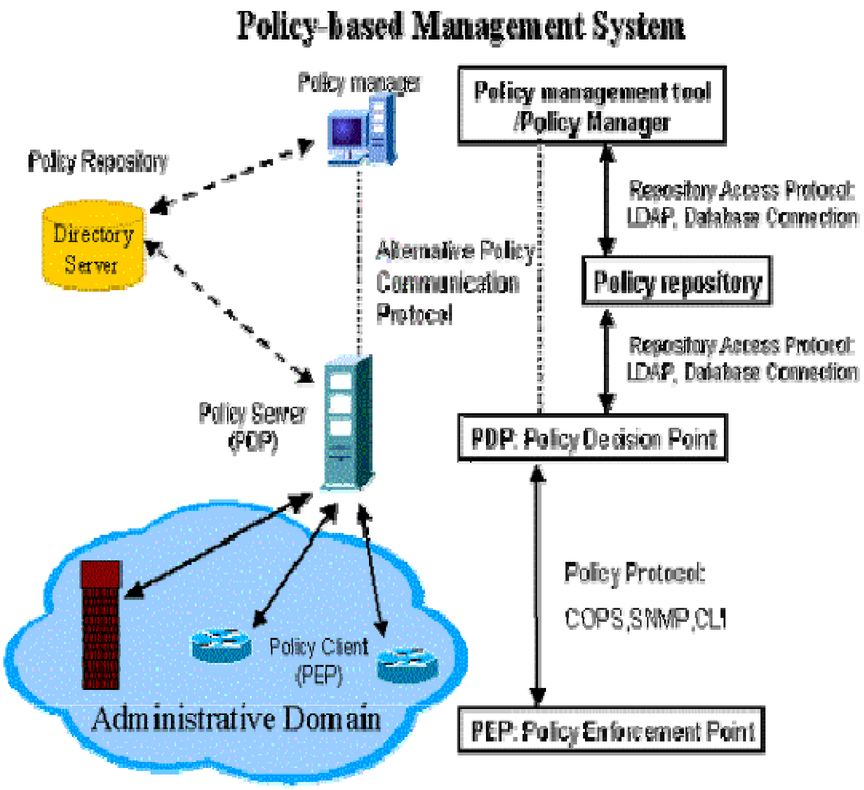

Figure 2. IETF policy framework

However, MPLS per se does not have QoS features nor mechanisms, but MPLS together with Differentiated Services (DiffServ) is the favored approach by the IETF for providing IP QoS. The second enhancement of the policy framework is dealing with network-level and service-level management in IP networks. Using MPLS networks, the notion of a Label Switched Path (LSP) brings network-level concepts into the frame-work which has not been dealt with in the device-level policy framework. Furthermore, using traffic engineered network, new kinds of IP services are possible. E.g., a service guaranteeing low packet loss probability can be offered using MPLS as mechanism to traffic engineer the IP network in a way that traffic is routed around hot spots [15].

4. Design a policy-based framework

4.1 Network topology

We deployed a MPLS network using Cisco router for the testing network plane. The infrastructure of this inters-AS MPLS VPN for Diffserv Qos testing was showing in Figure 3. MPLS network include P (Provider) router that is responsible for label swapping in MPLS backbone network 、PE (Provider Edge) router that is responsible for insert or pop label in the edge of MPLS network connecting with CE (Customer Edge) router which is in customer network and ASBR(Autonomous System Border Router) that is connected with other network service provider with MPLS network backbone. We use AS no. to distinguish with each other.

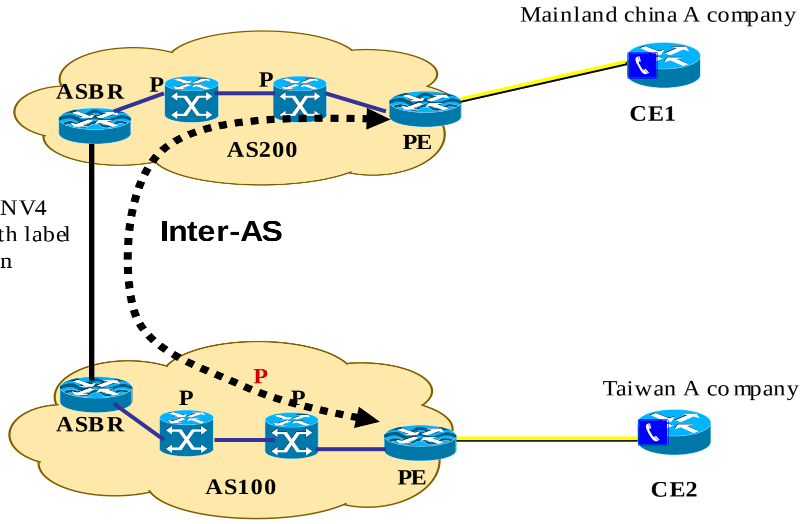

Figure 3. Diff-serv for Inter-AS MPLS VPN
. Figure 4 is a sample tunnel topology for unequal-cost load-sharing solution. We use Packet generator to create traffic flow into the MPLS backbone network to simulate real network condition.

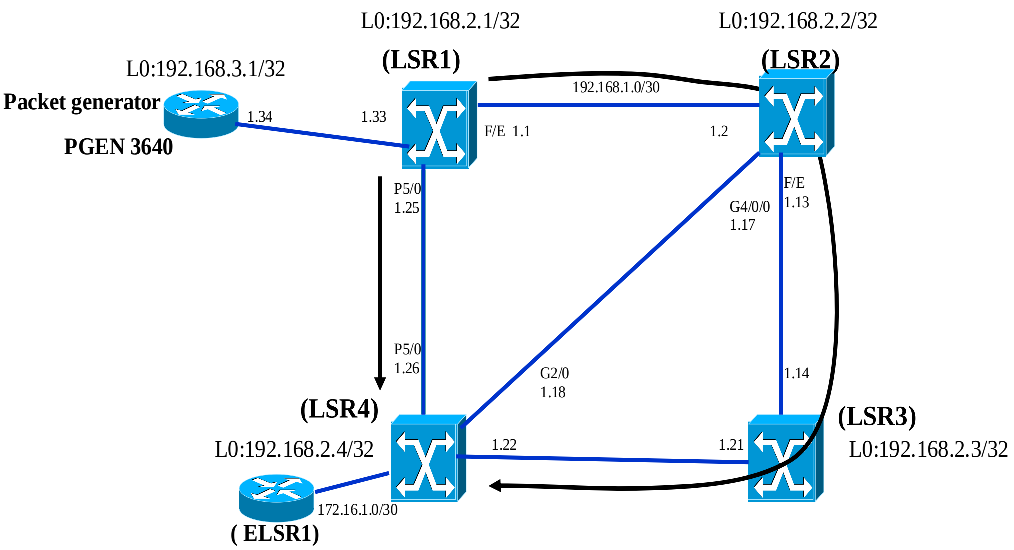

Figure 4. Inter-serv for MPLS network backbone load-sharing

4.2 Service-Level Agreement (SLA)

The contents of an SLA include the essential QoS-related parameters, including scope and flow identification, traffic conformance parameters, and service guarantees. More specifically, an SLA has the following fields: Physical Link, Topology, Attribute, Add service, FlowDes, Qos, and MPLS backbone network guarantees for

Performance Parameters, Service Schedule, and Reliability.

The scope of an SLA associated to a given service offering uniquely identifies the geographical and topological region over which the QoS of the IP service is to be enforced. An ingress (or egress) interface identifier should uniquely determine the boundary link or links as defined in [1] on which packets arrive/depart at the border of a DiffServ domain. This identifier may be an IP address, but it may also be determined by a layer two identifier in case of, say, Ethernet, or for unnumbered links like in, for example, Pointto- Point Protocol (PPP) access configurations.

5. Implementation

We developed a policy server prototype for the management of MPLS networks, in order to prove the feasibility of our architecture. The applicability to large MPLS/DiffServ networks has been shown by using Cisco router. However, at the current stage of the implementation, we can only show a working prototype proving the concept.

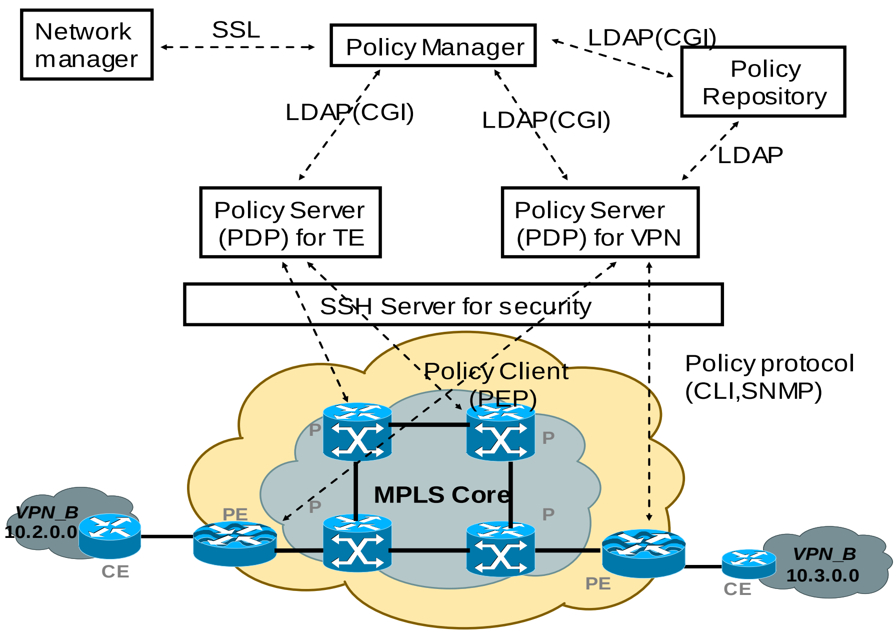

Figure 5. Policy-based Management framework

MPLS policy classes are converted into a LDAP directory schema. Furthermore, we built Cisco router to a MPLS network which offers MPLS functionality. The policy server and policy manager run on different PCs. The interface is implemented using a proprietary, COPS-like, text-based protocol between the real policy server and the Cisco router MPLS network using a TCP connection. All management agents send COPS-like messages to the real policy server. The messages from the Cisco router MPLS network to the policy server include always the network element's address and port number of the management agent.

5.1 Evaluation

We describe a prototypical implementation of a policy-based management system for MPLS Traffic Engineering, operating on MPLS network elements. Several experiment made in our test environment illustrate the general efficiency and feasibility of our architecture. For example, the setup of MPLS traffic engineering tunnel over four hops is performed in 2 seconds, and finally, MPLS traffic engineering tunnel deletion also lasts about 3 seconds, this data is calculated from Cisco router history log file and policy server history log file. Policy repository is using MySQL database software to establish, and policy server is using simulation software of telnet function like manual CLI (Command Line Interface) to send configuration information to Cisco router.

6. Conclusions

we propose a template for service-level agreement with a functional architecture for supporting the QoS required by contracted SLA. The management plane aspects of our architecture include SLA subscription, traffic forecasting, network dimensioning, and dynamic resource and route management. All of these are policy-driven.

We proposed a prototyping of a policy-based management system for MPLS Traffic Engineering, operating on MPLS network elements. Several experiments illustrate the general efficiency and feasibility of our architecture. Many of the functional blocks of this architectural model are also features of policy server the main difference being that a policy server is seen as driven purely by customer requests whereas in our approach, TE functions continually aim at optimizing the network configuration and its performance.

References

[1] S. Blake *et al.*, “An Architecture for Differentiated Services,” RFC 2475,.1998.

[2] P. Georgatsos *et al.*, “Technology Interoperation in ATM Networks: the REFORM System,” *IEEE Commun. Mag.*,vol. 37, no. 5, May 1999, pp. 112–18.

[3] D. Awduche *et al.*, “A Framework for Internet Traffic Engineering,” draft-ietf-tewg-framework-02.txt, work in progress, July 2000.

[4] E. Rosen, A. Viswanathan, and R. Callon, “Multi-protocol Label Switching Architecture,” RFC 3031, Jan. 2001.

[5] P. Aukia *et al.*, “RATES: A Server for MPLS Traffic Engineering,”*IEEE Network*, Mar./Apr. 2000.

[6] A. Feldmann *et al.*, “NetScope: Traffic Engineering for IP Networks,” *IEEE Network*, Mar./Apr. 2000.

[7] B. Teitelbaum, “Qbone Architecture (v1.0),” 1999; http://www.internet2.edu

[8] K. Nichols, V. Jacobson, and L. Zhang, “A Two-Bit Differentiated Services Architecture for the Internet,” RFC2638, July 1999.

[9] D. Goderis et al., “Service Level Specification Semantics and Parameters,” draft-tequila-sls-00.txt, work in progress, Nov. 2000.

[10] P. Flegkas et al., “On Policy-based Extensible Hierarchical Network Management in QoS-enabled IP Networks,” Proc. Policies for Dist. Sys. And Networks, , 2001.

[11] P. Aukia et al., “RATES: A Server for MPLS Traffic Engineering, ”IEEE Network, Mar./Apr. 2000.

[12] Dave Kosiur,Understanding Policy-Based Networking,2001, Willey Computer.

[13] Ivan Pepelnjak ,Jim Guichard, MPLS and VPN Architectures, 2001,Cisco Press.

[14] C.K.Wang,“Policy-Based Network Management”, WCC-ICCT, 2000.

[15] Jim Guichard,“A primer on Policy-based Network Management”, H-PCompany, 1999.