DESIGN AND IMPLEMENTATION OF AN INFORMATION MODEL FOR INTEGRATED CONFIGURATION AND PERFORMANCE MANAGEMENT OF MPLS-TE/VPN/QOS

Taesang Choi, Hyungseok Chung, Changhoon Kim and Taesoo Jeong
ETRI, {choits, chunghs, kimch, tsjeong@etri.re.kr}

Abstract: Multi Protocol Label Switching (MPLS) is generally considered as a mature technology. Many Internet Service Providers (ISPs) and telecommunication carriers have deployed it or are considering deploying it. An easy-to-use integrated management solution is requested by these ISPs. To realize a truly integrated management solution, a combined management information model is essential. In this paper, we propose an information model for integrated configuration and performance management of MPLS-Traffic Engineering (MPLS-TE)/VPN/QoS.

Key words: Information Model, MPLS-TE, MPLS-VPN, Diffserv, Configuration Management, Performance Management

1.   INTRODUCTION

As of today, Multi Protocol Label Switching (MPLS)[1] is considered as a mature technology. Many Internet Service Providers (ISPs) and telecommunication carriers have deployed it or are considering deploying it for various reasons: efficient usage of their valuable network and system resources and meeting customer's emerging service requirements such as provider managed IP virtual private networks (VPNs) and quality of service (QoS) guaranteed IP services for voice, video or mission critical applications.

One of the major requirements for a successful deployment is easy, efficient, scalable and reliable management of networks and services based on MPLS. . This includes automated provisioning, network and service performance monitoring, fault management and billing. And service providers want a more general management

solution and, thus, a common integrated information model for all these management functionalities is needed ever before.

In this paper, we propose an information model for integrated configuration and performance management of MPLS-TE/VPN/QoS. It is an OO-based abstract information model which means it is independent from the existing data models, encoding schemes and management protocols. In our proposal, we defined an information model by using Unified Modeling Language (UML) [2] which is used most widely to describe OO-based information models.

2. DESIGN OF THE INFORMATION MODEL

The main focus of the proposed information model is OO-based object model, protocol independency and integration of configuration and performance management. Our information model consists of four sub-models: one for MPLS-TE, MPLS-VPN, Diffserv[3]-based QoS, and Topology. Each sub-model is divided into configuration and performance parts. Topology sub-model is common to all three services. IP layer topology is a common denominator of MPLS-TE, MPLS-VPN and QoS topologies. Given the IP layer topology, MPLS-TE, MPLS-VPN or Diffserv topological information can be further added depending on an underlying network's capabilities. For example, if the underlying network supports MPLS-TE then both IP and MPLS-TE topological information is captured in the same topology information model. MPLS-TE sub-model models configuration and performance management information of MPLS-TE such as a traffic trunk, Label Switched Path (LSP) tunnel, LSP Path and traffic statistics of LSP tunnels. Similarly MPLS-VPN and QoS sub-models define conceptual management information and their relationship of respective services' configuration and performance functionalities. Due to the limited space, we only describe one sub-model: MPLS-TE.

2.1 Information Model for MPLS-TE

Figure 1 shows the information model for MPLS-TE configuration management. It shows required object classes and their relationships. There are three important classes, a traffic trunk, an LSP tunnel, and an LSP. The traffic trunk models the one defined by RFC2702, "The requirements for traffic engineering over MPLS" [4]. It represents an aggregate of customer traffic flows belonging to the same service class or classes. It can be mapped into zero or more LSP tunnels for load sharing purpose. Each LSP tunnel is mapped into one or more LSPs which are represented by dynamically calculated paths or explicitly specified paths. Other classes are tightly coupled with these main classes. For example, a traffic trunk (TtC) and LSP tunnel (LspTunnelC) need forwarding equivalence class (FEC) and a RSVP traffic profile (RsvpTp). FEC is a traffic classification filter and RsvpTP is a traffic profile attribute specification such as bandwidth, delay, jitter, etc.. Classes with letter "C" like TtC contain configuration and static information only. When a new class object is created, three objects (with letter "C", "M", and "S") are created at the same time and stores configuration-, performance-, and simulation-specific information separately for information consistency.
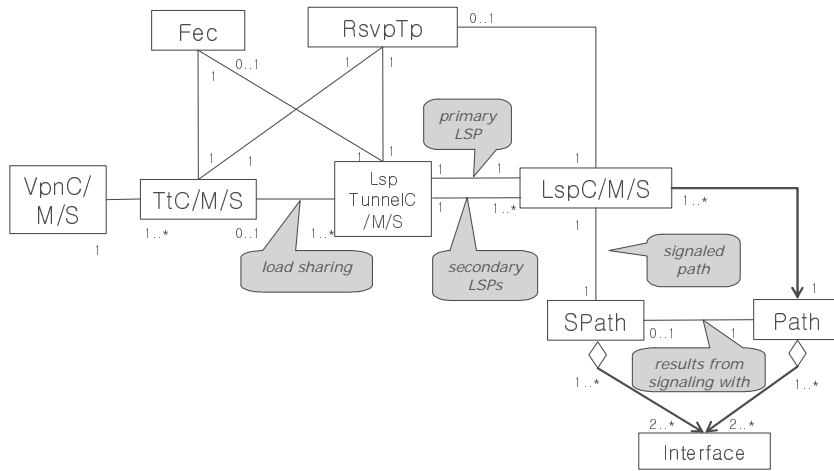
Figure 1. MPLS-TE Configuration Information Model

Classes with letter "M" denote that they are used for performance management. Traffic trunks, LSP tunnels and LSPs have their operational status information and statistics information such as packet per second every five minutes and bits per second every five minutes. Letter "S" stands for simulation. We identified some of the simulations which are very useful for the performance management. MPLS LSP path availability check, Node/Link failure, Traffic trunk and/or LSP tunnels attribute change and global optimization simulations are some of possible candidates. Simulations can be performed off-line with historical data acquired by off-line means. In such a situation, the simulation is typically considered to be a separate auxiliary mechanism to help performance management. In our proposed information model, we approached in a different way. These simulations are performed on-line with data collected and monitored live from the managed networks, which are stored in topology and performance sub-models. Since simulation results can modify the existing topological and performance information, separate place holders are needed. We modeled these "S" classes for that purpose.

The path availability check function provides an efficient way of simulating a setting up process of an LSP. The CSPF (Constrained Shortest Path First) algorithm, which is resident in our integrated management system, can compute the availability of a path of an LSP without actual enforcement. The server-based CSPF can also extend its scope to add additional constraints, e.g. actual usage instead of the required bandwidth of an LSP, besides what the online CSPF allows. This feature is one of the big advantages that an offline TE management system can provide. The LSP attribute modification simulation allows network managers to evaluate the side effects of LSP and VPN attribute modification. Modification of attributes ranges from simple single value change (e.g., the affinity value) to an entire path alteration for an LSP. This simulation helps the network managers create a detour route when a particular link is congested and see the link state changes in real-time. The link and node failure simulation depends on an online protection and recovery mechanism and visualizes its effects. Features like standby secondary paths, as well as explicit

or dynamically configured primary and secondary paths of LSPs are also recognized for this simulation. Depending on the situation, the simulation can just visualize the overall status of a newly optimized network status or visualize all the paths computed by the server's CSPF algorithm. The global optimization is performed by a customized algorithm based on linear programming (LP). The algorithm can find near optimal paths that satisfy a given traffic demand under some constraints, such as bandwidth, a maximum hop count, and a preferred or avoided node or link list. Our integrated management system generates a mixed integer programming formulation for a given optimization problem and solves it with a dedicated LP solver. The optimization result contains each LSP's routing paths and the traffic split ratio, in case an LSP requires multiple paths. For easy representation at network nodes, the split ratio is chosen among discrete values (0.1, 0.2 etc.). The globally optimized set of paths can then be applied to the MPLS networks that permit explicit path setup. For more details, please refer to [5].

3. IMPLEMENTATION

The proposed information model is implemented and used in the integrated configuration and performance management system called Wise<TE> [6]. We chose CORBA IDL as a language to specify this information model and implemented using C++. We have tested it over a network with 50 nodes and found out that it showed a good performance. Main concern is how the implementation can scale in terms of the number of CORBA objects (e.g., topology objects, LSP tunnel objects, MPVL-VPN l3vpn objects, etc.) as the number of nodes increase. Our implementation is running in a 500MHz dual CPU PC server with intel solaris 2.8. We are planning to test our system over a network with upto several hundreds nodes. At the same time, we are in the process of testing it in one of major ISPs in Korea.

REFERENCES

[1] E. Rosen, A. Viswanathan, R. Callon, "Multiprotocol Label Switching Architecture ", RFC3031, IETF, Jan. 2001.

[2] Object Management Group, "Unified Modeling Language (UML), Version 1.4", formal/2001-09-67, September 2001.

[3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services", RFC2475, IETF, December 1998.

[4] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, "Requirements for Traffic Engineering Over MPLS", RFC2702, IETF, September 1999.

[5] Y. Lee et al., "A Constrained Multipath Traffic Engineering Scheme for MPLS Networks," IEEE ICC 2002, New York, May 2002.

[6] TS Choi, SH Yoon, HS Chung, CH Kim, JS Park, BJ Lee, TS Jeong, "Wise<TE>: Traffic Engineering Server for a Large-Scale MPLS-based IP Network", NOMS 2002, April 2002.