A Hierarchical LSP Management Architecture for MPLS Traffic Engineering*

Daniel Won-Kyu Hong[1], Choong Seon Hong[2], and Dongsik Yun[1]

[1] Operations Support System Lab., R&D Group, KT
463-1 Jeonmin-Dong Yuseong-Gu, Daejeon 305-811 KOREA
{wkhong,dsyun}@kt.co.kr
[2] School of Electronics and Information, Kyung Hee Univerity
1 Seocheon Giheung Yongin, Gyeonggi 449-701 KOREA
cshong@khu.ac.kr

Abstract. In this paper, we propose a scalable Label Switched Path (LSP) management architecture for Multiprotocol Label Switching (MPLS) traffic engineering using a hierarchical network model. MPLS introduces the concept of a label hierarchy to support scalability by lessening the complexity of transit routers for the computation of complete routing tables. However, this hierarchical network model creates a critical disadvantage in providing a globally-optimal route because of topology abstraction or aggregation. This paper proposes a hierarchical routing scheme that can provide a globally-optimal route in hierarchical MPLS networks using propagation from the Condensed Subordinate Route Information (CSRI). CSRI is summarized route information among border nodes in the lower layer network and is reflected in the process of the LSP path computation in the higher layer network. We also propose the algorithms for generation of CSRI, for reflection CSRI to higher layer network topology, and for computation of an optimal LSP in a higher layer.

1 Introduction

The Internet is becoming an ideal platform to support modern communications including voice, data, and multimedia transmissions. However, standard IP routing protocols were developed on the basis of a connectionless model with routing decisions based on simple metrics, such as delay or hop count, which lead to the selection of shortest path routes [1,2,3]. Despite its ability to scale to very large networks, this approach provides only basic Quality of Service (QoS) capabilities, which are unable to provide scalable-service level agreements for bandwidth intensive applications in modern networks. Multi-protocol Label Switching (MPLS) extends IP destination-based routing protocols to provide new and scalable routing capabilities.

MPLS traffic engineering is inherently uses explicitly-routed paths. LSPs are created independently, specifying different paths based on user-defined policies;

* This research was supported by ITRC Project of MIC.

H.-K. Kahng and S. Goto (Eds.): ICOIN 2004, LNCS 3090, pp. 689–698, 2004.

however, this may require extensive operator intervention. RSVP and CR-LDP are two possible approaches to supply dynamic traffic engineering and QoS in MPLS [4,5,6]. Conversely, MPLS has been introduced into LSP hierarchy. LSP hierarchy is the idea that LSPs can be nested inside other LSPs, giving rise to an LSP hierarchy. This is achieved by considering an LSP as a link in the IS-IS or OSPF link state database [6], which is termed a layering concept in this paper. LSP hierarchy has been introduced to enhance scalability by reducing the complexity of transit routers from the composition of all network routing tables. Alternatively, a network can be logically or physically partitioned according to the service provider (SP) domain and the administrative domain within the same service provider, such as access and backbone networks, which will be defined as a partitioning concept in this paper. Layering and partitioning concepts are very useful when deploying a large-scaled network. However, these concepts bring a disadvantage in that they cannot provide a globally-optimal route in a hierarchical network model because of network topology abstraction and aggregation [8,9,10,11,12]. Consequently, we need a new scheme for guaranteeing the provisioning of a globally-optimal route in a hierarchical network model: the major goal of MPLS traffic engineering.

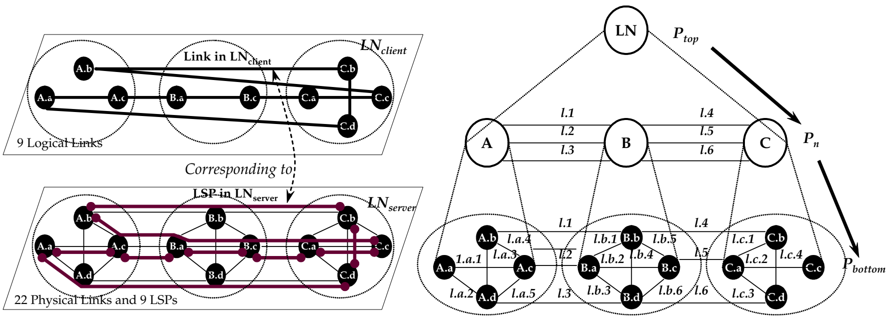
This paper proposes a scalable LSP management architecture that can provide a globally-optimal route in the hierarchical MPLS network. We will define the hierarchical network model applicable for a MPLS network. In addition, we will define the LSP management framework based on proposing a hierarchical network model for provisioning optimal LSPs in terms of MPLS traffic engineering. The paper is organized as follows: In section 2, we identify problems of the hierarchical model and define the hierarchical network model applicable for MPLS traffic engineering. In section 3, the LSP management framework for MPLS traffic engineering is discussed in detail. In section 4, we discuss performance issues of the proposed hierarchical LSP management framework. Finally, resulting conclusions are presented.

2 A Hierarchical Network Model

In this section, we discuss the generic hierarchical network model and identify problems of MPLS traffic engineering in a hierarchical network. In addition, we propose a new hierarchical network model that can solve the problems associated with layering and partitioning concepts.

2.1 Generic Hierarchical Network Model

A hierarchical network model is a traditional solution to the scaling problem [8]. The layer network (*LN*) represents a network boundary that can transfer certain kinds of network traffic without adaptation. Typical LNs can be IP, ATM, SDH, MPLS, et al. There can be client/server relationships among different LNs. Layer networks (*LNs*) are organized into different interconnected sub-networks (*SN*) called domains. An *SN* consists of multiple interconnected network nodes such



(a) Layering concept (b) Partitioning concept

Fig. 1. Layering and Partitioning Concepts

as IP routers, ATM switches, MPLS node, etc. An *SN* can be further partitioned until the lastly partitioned *SN* corresponds to a network node.

Fig. 1 shows layering and partitioning concepts. We apply the layering concept to the LSP hierarchy. As shown in Fig. 1 (a), the server layer network (LN_{server}) consists of 12 MPLS nodes, 22 physical links (Lp) and 9 LSPs. In LN_{server}, there are two kinds of LSPs: one is the end-to-end LSP (LSP_e) that can directly deliver customer MPLS packets, the other is the trunk LSP (LSP_t) that contains a number of subordinate LSPs. Fig. 1 (a) shows only LSP_ts. The LSP_ts correspond to the logical link (Ll) in the client layer network(LN_{client}). There is a *"corresponding to"* relationship between LSP_t in LN_{server} and Ll in LN_{client}.

An LN_{client} consists of a number of interconnected MPLS nodes that terminate the LSP_t at LN_{server}, and a number of Lls corresponding to the LSP_ts at LN_{server}. An LN_{client} is purely logical, whereas an LN_{server} is purely physical. Fig. 1 (b) shows the partitioning concept. An *LN* can be partitioned according to the different service provider domains or the different administrative domains within the same service provider domain. There may also be *N*-partitioning levels. The top partitioning level (P_{top}) represents the *LN* itself. The lowest partitioning level (P_{bottom}) corresponds to the entire network topology of an *LN*. There can be a number of subsequent partitioning levels between P_{top} and P_{bottom}, determined by the administrative policy of each network service provider. In Fig. 1 (b), the LN (P_{top}) is composed of three *SNs (A, B,* and *C)* and six interconnecting links (*l.1-l.6*), which is the first partitioning level. At the second partitioning level, there are three individual *SNs* of *A, B,* and *C.* The *SN A* consists of four nodes *(A.a, A.b, A.c,* and *A.d)* and five links *(l.a.1, l.a.2, l.a.3, l.a.4,* and *l.a.5).* In addition, all *SNs* have a number of border nodes ($N_{borders}$) with connectivity to other *SNs*, for example: *A.b, A.c,* and *A.d* in the case of *SN A*, *B.a, B.b, B.c,* and *D.d* in the case of *SN B*, and *C.a, C.b,* and *C.d* in the case of *SN C*.

2.2 The Problems of a Hierarchical Network Model in Terms of MPLS Traffic Engineering

The partitioning concept is very useful for deploying a large-scaled MPLS network reflecting the different kinds of administrative policies. However, we cannot provide a globally-optimal route in a hierarchical MPLS network that is deployed based on a partitioning concept. The P_{top} only shows its subordinate network topology. The intermediate partitioning level (P_n) hides P_{bottom} from P_{top}. Therefore, P_{top} finds route with P_n. In addition, P_n finds route with P_{bottom}. For example, if we find a route from $A.a$ to $C.c$ using the shortest-path first routing algorithm, P_{top} selects a route traversing *l.1* and *l.4*, *(A-l1-B-l4-C)*, which can be a reasonable path in terms of P_{top} because the hop counts of all possible routes are the same. Subsequently, each *SN* of *A, B,* and *C* finds an optimal route with their partitioned network topology as shown in Fig. 1 (b). *SN A* selects *(A.a-l.a.2-A.d-l.a.3-A.b)*, *SN B* selects *(B.b)*, and *SN C* selects *(C.b-l.c.3-C.a-l.c.2-C.c)*. As a result of hierarchical routing, the selected route is *(A.a-l.a.2-A.d-l.a.3-A.b-l.1-B.b-l.4-C.b-l.c.3-C.a-l.c.2-C.c)*, which traverses six hops. However, there is a more optimal route than the selected route, which traverses five hops, *(A.a-l.a.1-A.c-l.2-B.a-l.b.s-B.c-l.5-C.a-l.c.2-C.c)*. If P_{top} selects *l.2* and *1.5*, then we can find the most optimal route in a hierarchical environment. In contrast, layering concept gives us an unfavorable side effect whereby the LN_{client} selects an explicit route for LSP configuration with its network topology. The LN_{client} has the highest probability of selecting the Ll , which has a longer transit delay than the others because LN_{client} only has the link attributes of Ll but with the path attributes of LSP_t at LN_{server}.

2.3 A New Hierarchical Network Model

In this paper, we propose a hierarchical network model that highlights the advantages and solves the disadvantages in hierarchical networks. First, we propose the propagation of Condensed Subordinate Route Information (CSRI) to solve problems caused by the partitioning concept. CSRI is a set of summarized routes to be propagated from the subordinate partitioning level to the upper partitioning level.

Each SN computes all possible routes (R) using partitioned network topology. We use the Dijstra algorithm for computation of Rs. There are border nodes (N_b) within the partitioned network topology; for example, *A.b, A.c,* and *A.d* in Fig. 2. If there are n border nodes, the number of SCRI entries can be $n(n-1)/2$. SCRI can be defined as $SCRI = (h, d, b, r)$, where h is hop count, d is accumulated delay, b is available bandwidth, and r is extent or route availability ($r = yes/no$). First, we computed all possible routes between two border nodes: (n_{source}) and ($n_{destination}$). By computing routes between n_{source} and ndestionation, we determined the metrics of R, including hop count ($R(H)$), delay ($R(D)$), bandwidth ($R(B)$), and availability ($R(A)$) as shown in Fig. 3. H is the number of links traversing the computed route R, $R(D)$ is the accumulated delay of links traversing the computed route R, $R(B)$ is the smallest bandwidth

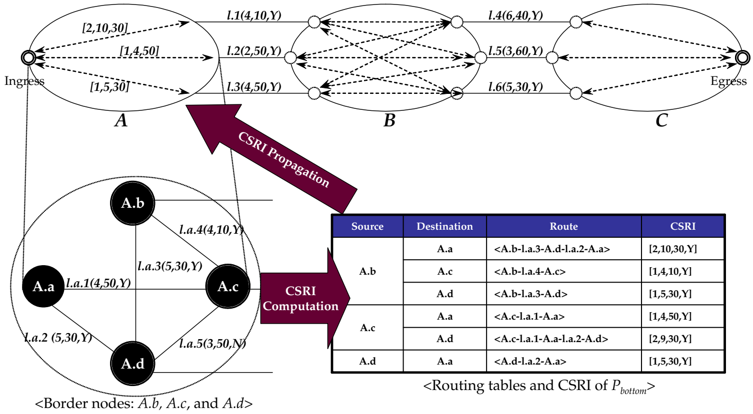


Fig. 2. CSRI propagation model

Let N_b is the list of border node within the partitioned network topology
Let $R=\{(n,l)\}$ is the computed route, where n is a node and l is an link.
Let $l=(D,B_{ava},A)$ is the link, where D is a transmission delay of the link, B_{ava} is the available bandwidth of the link, and A is the availability of the link

For $\forall n_{source} \in N_b$ do {

for $\forall n_{destination} \in (N-n_{source})$ do {

i= 0;

Compute all possible route *(R)* between n_{source} and $n_{destination}$ with the following rules:

***R(H)* is the number of links traversing the computed R_i**

***R(D)* is the accumulated delay of links traversing the computed R_i**

***R(B)* is the smallest bandwidth of link among the bandwidths of links traversing the computed R_i**

***R(R)* is the reachability of the computed R_i**

if there is a computed route, then *i++;*

}

// Computation of CSRI

$CSRI(n_{source},n_{destination})(h,d,b,r) = (min(\Sigma R(H)), min(\Sigma R(D)), max(\Sigma R(B)), \Sigma R(A)== yes);$

}

Fig. 3. CSRI computation algorithm

of the link among those traversing the computed route R, and $R(A)$ can be "*no*" if one of links traversing the computed route R is "*no*". In order to compute, we used the Dijstra algorithm.

After computing all possible routes between n_{source} and $n_{destination}$, we determined the CSRI metrics according to an administrative policy. There can be one or more possible routes between an identical n_{source} and ndestiation. CSRI metrics can be of the first priority route. Then, how can we select the first priority route among computed routes? In general, we selected aroute having the smallest hop count, smallest accumulated delay, largest available bandwidth, and a accessible route as the first priority route. However, if the service provider wishes to deploy a delay sensitive network, then we must select the route having the least accumulated delay over other metrics as first priority. As we selected the most optimal route among all possible routes between n_{source} and $n_{destination}$, and defined the metrics of the first priority route as CSRI metrics, we dramatically

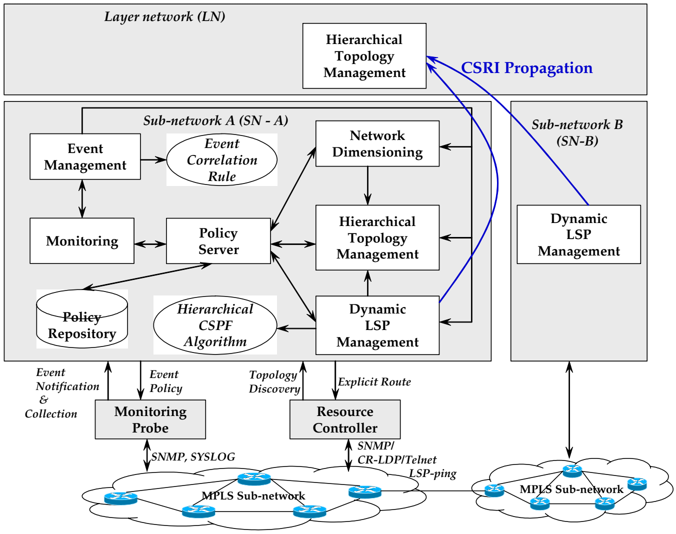


Fig. 4. LSP management architecture

reduced the number of CSRI entries to be propagated to the upper partitioning level. This resulted in topology aggregation or filtering. After selection of CSRI entries, we propagated CSRI to the upper partitioning level as shown in Fig. 2. Propagated CSRI is reflected into the internal topology of the upper partitioning *SN*. Next, we selected the globally-optimal route reflecting CSRI at the N_{top}. If we did not reflect CSRI at N_{top}, the computation of route at N_{top} would result in an unfavorable route. Thus, we provided a globally-optimal route in a hierarchical network model using the concept of CSRI propagation, which makes a network service provider supply traffic-engineered LSP in a hierarchical MPLS network.

3 LSP Management Architecture

In the previous section, we proposed a hierarchical network model to guarantee the provision of globally-optimal traffic engineered LSP in a hierarchical MPLS network. In this section, we propose LSP management architecture from the perspective of service and network management.

There are four broad areas in LSP Management Architectur: network device control, network configuration, event & fault management, and policy management areas. Each sub-network has these four management functions. The device control mediator (DCM) takes the roles of device configuration and monitoring. The resource controller (RC) configures network devices using SNMP, Telnet, and CR-LDP; it also checks the validity of a configured LSP using LSP-Ping.

The monitoring probe (MP) collects myriad events generated from network devices and propagates the events to the event management module.

Policy repository (PR) has rules provisioning LSP under a hierarchical environment. These include rule-based constraints on optimization processes, alarm triggering, and propagation. The network-dimensioning module collects periodic MPLS traffic patterns and analyzes the gathered traffic. According to traffic analysis, the network-dimensioning module configures the hierarchical MPLS network topology as described in section 2.3. It determines the affinity, color, and capacity of all links, reconfigures existing LSPs, and decides protection schemes for each LSP such as 1:1 protection, 1:N protection, N:M protection, or rerouting.

The dynamic LSP management module configures LSPs using the hierarchical CSPF algorithm. This module receives LSP configuration requests from the network-dimensioning module in terms of network planning and the network operator in terms of the on-demand LSP configuration, including MPLS-VPN configuration. In addition, this module maintains numerous provisioned LSPs. In order to support the optimal path provision in a hierarchical MPLS network, the LSP management module calculates CSRI entries and propagates them to the hierarchical topology management module in the upper-layer management system.

The hierarchical topology management module maintains a hierarchical MPLS network topology provisioned and planned by the network-dimensioning module. Network topology is used to calculate an optimal path for LSP provision. In addition, network status is dynamically reflected into the network topology in order to find an optimal path taking into account the current network status. Subsequently, the hierarchical topology management module dynamically changes the status of network topology according to innumerable network events originating in the event-management module. While maintaining hierarchical network topology, the topology management module receives CSRI entries from a LSP management model at the subordinate management system, and reflects received CSRI to the internal cost of sub-networks.

Thus, propagation of CSRI entries between the LSP management model at the subordinate management system and the configuration management model at the upper management system provides a globally-optimal route in a hierarchical network environment.

4 Performance Issues

Our LSP management architecture can provide aglobally-optimal route in hierarchical MPLS network environments. In this paper, we measured the LSP configuration performance in terms of route computation and resource utilization comparing our approach to so-called "CSRI propagation approach" with two other approaches. The two other approaches are as follows: one is to find a route with flat network topology with no hierarchy, a so-called flat topology approach; and the other is to find a route without CSRI propagation in a hierarchical network environment, a so-called "no CSRI propagation approach". Fig. 5

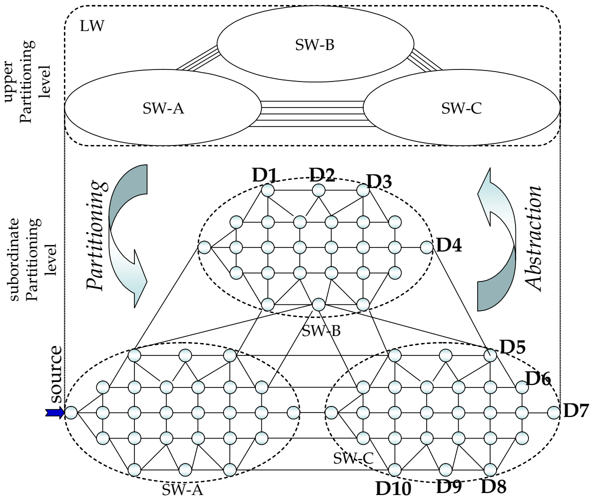
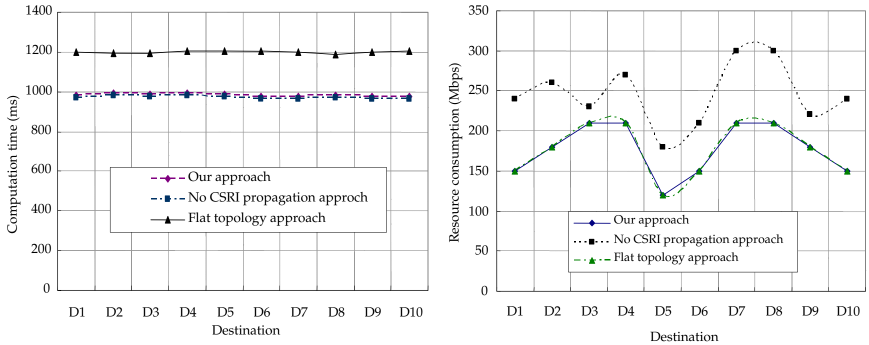


Fig. 5. A simplified test topology

shows a simplified network topology for simulation. There are two partitioning levels. Layer network (LN) consists of three sub-networks: $SW - A$, $SW - B$ and $SW - C$, and ten links connecting the tree sub-networks. In contrast, each sub-network has 26 nodes and 51 links connecting the nodes.

We assume that all links connecting nodes and sub-networks has the same available bandwidth and transit delay. We found a route between the fixed source and ten variant destinations as shown in Fig. 5. We calculated the path at 50 times between the same source and destination. Fig. 6 shows the average performance of 50 attempts. In terms of path computation performance, the "no CSRI propagation approach" shows higher performance than the other two approaches because it finds a route with the highly-abstract network topology of three sub-networks and 13 links. However, our approach showed slight performance degradation compared to the "no CSRI propagation approach". The performance degradation of our approach compared to the "no CSRI propagation approach" stems from CSRI propagation overheads from the subordinate (sub-network) to the upper level (layer network). Conversely, the flat topology approach showed the worst performance because it computed an optimal path with the entire network topology composed of 78 nodes and 166 links. In the case of "no CSRI propagation approach" and our approach, each computed the optimal path in a distributed and concurrent way. That is to say, each sub-network computes its own route with the partitioned network topology composed of 26 nodes and 51 links.

In this simulation, we tried to compute a path for LSP provisioning with 10Mbps between the fixed source and ten variable destinations as shown in Fig. 5.



(a) Path computation (b) Resource utilization

Fig. 6. Performance comparision

From the perspective of network resource utilization, our approach showed the highest performance of all approaches including the "no CSRI propagation approach" and flat topology approach as shown in Fig. 6 (b). By propagating CSRI from the subordinate to the upper level, and reflecting CSRI into the internal cost of node at the upper level, thereby showing the globally-optimal route in a hierarchical network environment. Even though the "no CSRI propagation approach" finds a route with abstracted network topology at the upper level, it cannot guarantee optimal route provision in a hierarchical network environment. Therefore, the resource utilization of the "no CSRI propagation approach" performed the worst among our approach and the flat topology approach as shown in Fig. 6 (b).

The flat topology approach showed nearly the same performance as our approach in terms of network resource utilization; this is because it finds a route with full network topology without abstraction. Taking network resource utilization and path computation speed into account, our approach is the most reasonable compared to the flat topology approach and the "no CSRI propagation approach" in a hierarchical MPLS network environment.

5 Concluding Remarks

In this paper, we proposed a scaleable LSP management architecture that can provide a globally-optimal route under the hierarchical MPLS network. We defined a hierarchical MPLS network model with layering and partitioning concepts. By introducing CSRI propagation from the subordinate partitioning level to the upper partitioning level, we can guarantee a globally-optimal LSP provision under the hierarchical network environment, which was impossible in the hierarchical network environment because of topology abstraction. In addition, this paper proposed an LSP management architecture that is applicable for LSP provision in a hierarchical MPLS network environment.

Through simulation, we compared our scheme with the approaches of flat topology and "no CSRI propagation" under a somewhat complex topology of 78 nodes and 166 links. As a result of this simulation, we proved that our approach is the most reasonable one when compared to the other two when taking path computation speed and resource utilization into account.

References

[1] R. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture, "RFC3031, January 2001.
[2] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and McManis, "Requirements for Traffic Engineering over MPLS, "RFC2702, September 1999.
[3] S. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and Principles of Internet Traffic Engineering," RFC3272, May 2002.
[4] X. Xiao and L. M. Ni, "Internet QoS: A Big Picture," IEEE Network, March/April 1999.
[5] X. Xiao, A. Hannan, B. Bailey, and L. M. Ni, "Traffic Engineering with MPLS in the Internet," IEEE Network, March 2000.
[6] A. Banerjee, J. Darke, J. P. Lang, B. Turner, K. Kompella, and Y. Rekhter, "Generalized Multiprotocol Label Switching: An Overview of Routing and Management Enhancements," IEEE Communications Magazine, pp.144-150, January 2001.
[7] Gerald R. Ash, "Performance evaluation of QoS-routing methods for IP-based multiservice networks", Computer Communications Vol 26, pp.817-833, 2003.
[8] M. El-Darieby, D. C. Petriu, and J. Rolia, "A Hierarchical Distributed Protocol for MPLS path creation", Proc. of 7th IEEE International Symposium on Computers and Communications, pp.920-926, July 2002.
[9] E. Felstine and R. Cohen, "On the Distribution of Routing Computation in Hierarchical ATM Networks," IEEE Transactions on Networking, Vol. 7. No. 6, December 1999.
[10] B. Awerbuch, Y. Du, and Y.Shavitt, "The Effect of Network Hierarchy Structure on Performance of ATM PNNI Hierarchical Routing," Computer Communications Journal, vol 23, pp.980-986, 2000.
[11] A. Iwata and N. Fujita, "A Hierarchical Multilayer QoS Routing System with Dynamic SLA Management," IEEE Journal on Selected Areas in Communication, Vol. 18, No. 12, pp.2603-2616, December 2000.
[12] Y. Qin, L. Mason, and K. Jia, "Study on a Joint Multiple Layer Restoration Scheme for IP over WDM Networks," IEEE Network, pp.43-48, March/April 2003.