8

ADAPTIVE LSP

8.1 INTRODUCTION

As the Internet has evolved from its research origins into a popular consumer technology, network resource management has become a main problem for service providers. It was thought that the solution to the problem would be new technologies capable of providing sufficient network resources like cheap memory, high-speed links and high-speed processors. Though these improvements contribute significantly to enhancing traffic performance, they do not solve the problem of optimal use of network resources. One of the most important functions performed by the Internet is the routing of traffic from ingress nodes to egress nodes. The most commonly used shortest path routing protocol chooses as a preference the shortest link to forward packets. This ignores performance information which forces communication over excessively long or overloaded links leading to non-optimal path selection or unbalanced network load situations. Therefore, one of the main tasks to be performed by Internet Traffic Engineering (TE) is the control and optimization of routing functions to forward

traffic through the network in the most effective way [ACE+02] [AMA+99]. Thus, the main focus of Internet TE is to facilitate efficient and reliable network operations while simultaneously optimizing network resource utilization and traffic performance.

The optimization objective of Internet traffic engineering should be viewed as a continual and iterative process of network resource utilization improvement. Different networks may have different optimization objectives depending on the network utility models. However, in general, TE optimization focuses on network control regardless of the specific optimization objectives. One major challenge of Internet TE is the realization of automated control capabilities that adapt to significant changes quickly and cost effectively, while still maintaining stability.

MPLS traffic engineering provides an integrated approach to TE. It routes traffic flows across a network based on the resources the traffic flow requires and the resources available in the network. It also employs "constraint-based routing" in which the path or Label Switching Path (LSP) for a traffic flow is the shortest path that meets the resource requirements (constraints) of the traffic flow.

The Label Distribution Protocol (LDP) is in charge of setting up an LSP with a given maximum bandwidth. While the demand bandwidth of the aggregated flows in a particular LSP is less than or equal to the maximum bandwidth assigned to this LSP, the Label Edge Router (LER) continues sending traffic to the established LSP without any problem. The problem arises when the bandwidth demand for the aggregated flows becomes greater than the maximum assigned capacity. Obviously, the traffic demand changes over time but the topology and routing configuration cannot be changed as rapidly. This causes the network topology and routing configuration to become sub-optimal over time, which may result in persistent congestion problems on the LSP. The other issue occurs when the reservable bandwidth in a link on the shortest path (optimal connection) does not meet the bandwidth constraint for the new demands. This situation obliges the routing protocols to select a non-optimal LSP. Note that the reservable bandwidth of a link is equal to its capacity minus the

total bandwidth reserved by LSPs traversing the link. It does not depend on the actual amount of available bandwidth on that link.

In the literature there are different approaches suggested to tackle these network problems. We summarize them as follows.

1. Traffic losses.
2. Create new LSP with more maximum BW (BWmax) and reroute all aggregated traffic on it.
3. Use traffic engineering to split traffic onto a new LSP.
4. Modify the LSP bandwidth, if possible [ALAS+02].

The first option simply decides to drop the excess traffic to control the congestion in the network. This can't be applied any traffic with QoS requirements. The solution is simple, but it is not appropriate for critical traffic.

Options (2) and (3) introduce additional overhead by extra signaling processes to establish the new LSP, but they are transparent to traffic. Option 3 in particular has a problem with regard to network scalability, which is inversely proportional to the number of labels used by an LER to forward the same amount of traffic.

The last option, proposed by Ash et al. [ALAS+02], modifies the bandwidth of an established LSP using CR-LDP. It is transparent to traffic. Here too extra signaling for the additional bandwidth request is required.

All the above mentioned options address the problem of satisfying only the traffic requirement (additional bandwidth demand) triggered by some congestion problems on the network. Other aspects of performance and resource optimization that are not considered are i) to find an optimal LSP and reroute the traffic when there is traffic reduction, and ii) to reroute traffic from a non-optimal LSP to a better LSP when a previously established LSP is released.

The objective of this proposal is to contribute significantly to the improvement of MPLS traffic engineering considering the two performance aspects mentioned above.

8.2 RELATED WORK

In [AMA+99] the authors present a set of requirements for traffic engineering over MPLS, identifying the functional capabilities required to implement policies that facilitate TE in an MPLS domain. They classify the TE performance objectives mainly in two groups: traffic oriented and resource oriented. The first strives to enhance the QoS of traffic streams (packet loss, delay, delay variation, and goodput). The second deals with optimization and efficient network resource allocation and utilization. In [Awd99] the author defines the basic components of the MPLS traffic engineering model: path management, traffic assignment, network state information dissemination and network management.

The MPLS adaptive traffic engineering (MATE) presented in [EJLW01] addresses the network congestion problems using a multipath adaptive traffic engineering mechanism. The mechanism assumes that several explicit LSPs are set between an ingress and an egress node using a standard protocol such as CR-LDP [JAC+02]or RSVP-TE [ABG+01], or configured manually. The proposed adaptive TE mechanism uses probe packets to obtain LSP statistics such as packet delay and packet losses in order to shift traffic among LSPs. Here we clearly see that the MATE mechanism is not capable of modifying the LSP bandwidth to accommodate additional demands when all established LSPs reach their maximum reserved bandwidth. At the same time, the typical range between end nodes proposed in the MATE operational settings (from two to five explicit parallel LSPs) continues reserving the same amount of maximum bandwidth even if the traffic decreases drastically in all LSPs.

In [Swa99] two further optimization strategies are suggested. The first uses multiple LSPs to each destination - like MATE - to balance the load. But, instead of sending a probe packet to monitor the utilization of each LSP, it uses link utilization informa-

tion by extending ISIS or OSPF. The second approach attempts to auto-adjust the bandwidth based on the real usage of an LSP.

In the proposal of Ash et al. [ALAS+02] the authors address the problem related to additional bandwidth requirements for the traffic carried on an LSP. The work presents an approach modifying the bandwidth of an established LSP using CR-LDP without service interruption. The proposed mechanism not only addresses the increase of bandwidth to accommodate the new bandwidth demand, it also includes the possibility to decrease LSP bandwidth when the traffic on the LSP has decreased. In this case, their method releases the delta (difference) bandwidth (ΔBW) and continues using the same LSP.

It is clear that some proposed mechanisms try to solve the congestion problems while others try to accommodate additional bandwidth demands. However, they don't cover the re-optimization of the previously established LSP by rerouting to optimal paths after significant changes in the network occur, such as a reduction of traffic on an LSP or the release of an LSP.

8.3 PROBLEM FORMULATION

We will use an example in a simple scenario in order to illustrate the problem to be solved. In Figure 8.1 we present a simple MPLS network scenario formed by four LERs as edge routers and six intermediate or transit LSRs. We also have four Autonomous Systems (AS) A, B, C and D connected to the MPLS network. In this example we establish the full mesh connection between these four LERs, and we analyze the operation for optimal LSPs and non-optimal LSPs to see the impact on resource utilization.

Building the full mesh of LSPs according to the shortest path (optimal) gives the configuration depicted in Table 8.1.

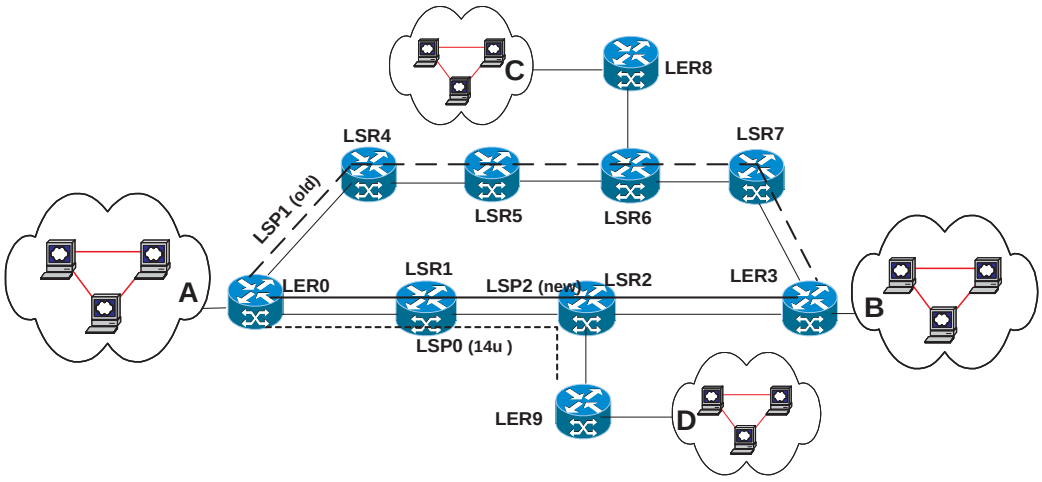

Figure 8.1 Scenario

Note that the maximum number of LSPs that can be established in the network is n*(n-1), where n is the number of LERs. In our case, as there are four LERs the total number of LSPs will be 4*(4-1) =12.

Building the full mesh of LSPs according to the non-optimal (non-shortest) path gives the following configuration. (Table 8.2.)

	A	B	C	D
A	——	0-1-2-3	0-4-5-6-8	0-1-2-9
B	3-2-1-0	——	3-7-6-8	3-2-9
C	8-6-5-4-0	8-6-7-3	——	8-6-7-3-2-9
D	9-2-1-0	9-2-3	9-2-3-7-6-8	——

Table 8.1 Full mesh optimal connection using shortest path algorithm

	A	B	C	D
A	——	0-4-5-6-7-3	0-1-2-3-7-6-8	0-4-5-6-7-3-2-9
B	3-7-6-5-4-0	——	3-2-1-0-4-5-6-8	3-7-6-5-4-0-1-2-9
C	8-6-7-3-2-1-0	8-6-5-4-0-1-2-3	——	8-6-5-4-0-1-2-9
D	9-2-3-7-6-5-4-0	9-2-1-0-4-5-6-7-3	9-2-1-0-4-5-6-8	——

Table 8.2 Full mesh with non-optimal connection

Table 8.3 presents a comparison of the number of LSPs in each link when non-optimal routing is used with respect to the optimal (shortest path) routing.

		Number of LSP per link	
Links	Number of links	Optimal	Non-optimal
0-4, 4-5, 5-6	3	2	10
0-1, 1-2, 6-7, 7-3	4	4	8
2-3, 2-9, 6-8	3	6	6

Table 8.3 Comparison table for fully optimal and non-optimal LSP connection

In the first row of the table we can observe that three network links (0-4, 4-5, 5-6) are shared by 10 LSPs for non-optimal routing, reporting the maximum number of LSPs per link. In the case of optimal routing these three links are shared by only 2 LSPs. The last row reports the maximum LSPs per link for optimal routing, which is 6 LSPs. Those links that are shared by the highest number of LSPs are to be considered to be "critical links" in the network [KL00].

Following this example and Table 8.3 we find that for non-optimal cases there are 10 LSPs in the "critical links", while for the optimal case there are 6 LSPs in the "critical links". Considering all links to be identical, with the bandwidth capacity of

C (link capacity), and all LSP have the same bandwidth assigned, the maximum link bottleneck in the network for optimal routing is equal to C/6 and for the non-optimal is C/10. As C/6 > C/10 we get better network resource utilization for the optimal rerouting. This condition causes us to look for a mechanism for rerouting non-optimal LSPs.

The second aspect we want to illustrate in this section is an example of the operation for increasing the bandwidth of an LSP. The scenario is the one depicted in Figure 8.1. We assume that all links have the same delay and a link capacity of 20 units (C=20).

We define a flow f(i,AS) as the flow number i from autonomous system AS. A Forward Equivalence Class (FEC) Fi corresponds to LERi as the destination node (egress LER) to leave the network.

Consider also that the path LER9-LSR2-LSR1-LER0 is occupied by flows from D to A, with demand for 14 units bandwidth forming the LSP0 for packets classified by LER9 as FEC F0. This path is formed by 3 links, and the cost associated with it is 3.

Now, the available link capacity for the path between LER0 and LER3 through LER0-LSR1-LSR2-LER3, with cost 3, is 6 units. And for the same path through LER0-LSR4-LSR5-LSR6-LSR7-LER3, with cost 5, the available link capacity is 20 units. This situation is common in real networks because the shortest paths (path with less cost) are the preferred paths to be selected by routing protocols.

Suppose that now A sends a flow f(1, A) to B with 10 units bandwidth demand. According to the MPLS architecture the LER0 associates the f(1,A) to the FEC F3 (i.e., LER3 is the destination node to leave the network), and after, it sends the label request message with 10 units of bandwidth to the downstream LSRs. During this process there is not sufficient bandwidth to accommodate the traffic through path LER0-LSR1-LSR2-LER3 (it has only 6 units left). So, the attempt to establish

an LSP using this link for the request is rejected. On the other hand, the downstream LSRs through the path LER0-LSR4-LSR5-LSR6-LSR7-LER3 have 20 units of bandwidth available and accept the request and map the corresponding label to F3. When the label mapping message is received by LER0 the establishment of LSP1 is concluded.

Following the example, suppose now that after the establishment of LSP1 for flows classified by ingress LER0 as F3, A sends a flow f(2,A) to B with 2.5 units of bandwidth demand. This flow has as destination LER3. This implies that LER0 will classify it as F3 and will assign the same label as for f(1,A) and forward it through LSP1. Assume also that after this process A sends a new flow f(3,A) to B with 3 units of bandwidth demand. This flow also receives the same treatment by LER0. This situation increases the bandwidth usage of LSP1 to 15.5 units. The bandwidth of the LSP accommodates the new requirements as new flows are aggregated.

Now we illustrate how non-optimal routing may lead to blocking new requests. Continuing with the previous example, suppose that C attempts to send a flow f(1,C) to D with 6 units. Its request will be rejected by a downstream LSR (LSR6) due to the lack of available bandwidth on the outgoing link (because LSR6-LSR7 has only 4.5 units available), resulting in the rejection of this request producing the blocking problem.

Another aspect of the desired behavior of the network is the ability to reroute current LSPs in order to evolve towards a more optimal routing configuration.

Using the same example situation, assume that after a certain time the flow f(1,A) ceases. Flows f(2,A) with 2.5 units and f(3,A) with 3 units on the LSP1 remain (in total, the used bandwidth is now 5.5 units). Now there is enough available bandwidth through path LER0-LSR1-LSR2-LER3 (6 units) to accommodate these flows (5.5 units). And we believe it is better to forward the remaining aggregated traffic from A to B through LSP2 instead of continuing to do it via LSP1. Doing so, we will be able to dynamically manage network resource utilization, and at the same time reduce the

delay that packets experience by using the path with cost 5 (5 links) instead of cost 3 (3 links). There are many proposals addressing fast rerouting of LSPs without service interruption, so that the rerouting is not a major issue.

Finally, consider the case when all link capacities for low cost paths (optimal LSPs) are occupied by traffic with the same priority. In this case even using Ash's proposal it is impossible to modify (increase) the LSP bandwidth due to the link capacity being fully used. As a result the incoming traffic is forwarded over a high cost LSP. Suppose that after a while, the traffic over the low cost LSP ceases (the associated LSP is released). In this situation if we continue sending the traffic through the non-optimal (high cost) LSP, evidently we are wasting valuable network resources.

For example, suppose that when LSP1 reaches 15.5 units of bandwidth usage, after the aggregation of three flows with 10, 2.5 and 3 units, LSP0 is released. In this condition we are able to reroute the traffic from LSP1 (15.5 units) to an LSP that can be established through path LER0-LSR1-LSR2-LER3 with a capacity of 20 units. Note that although the bandwidth usage (BWu) is not less than the assigned bandwidth threshold (BWt), using this mechanism we are able to reroute the traffic from a non-optimal LSP to the optimal LSP, improving the overall performance of the MPLS network.

Note that the modification of LSP bandwidth proposed in [ALAS+02] also includes the possibility to decrease the LSP bandwidth when the aggregated traffic has decreased. In this case their method releases the bandwidth equal to the difference of bandwidth between current and previous aggregated flows (delta bandwidth) and continues using the same LSP.

8.4 ADAPTIVE LSP ROUTING

In this chapter we propose an additional functionality to the edge LSRs (LERs) introducing new criteria to overcome the problem derived from non-optimal routing

at LSP setup time and provide better performance and network resource utilization to MPLS based networks.

The ingress LER must store bandwidth requests (demand) and dynamically monitor the LSP bandwidth usage compared with the assigned threshold value. At the same time it watches for released LSPs on low cost paths to transfer the same priority traffic from high cost LSPs.

After the establishment of an LSP, the ingress LER continues forwarding packets as per MPLS architecture procedures. Our mechanism starts by storing the information of the LSPs initial aggregated bandwidth demand (BWid) in the LER. Then the LER starts to monitor the aggregate bandwidth usage (BWu). This is possible because an LER both establishes the LSP and forwards the traffic into it, so all information needed for our proposal is readily available within the LER. If BWu remains above the threshold value (BWt) no action will take place. The threshold is defined to be some reasonable percentage of the initially allocated aggregated bandwidth (i.e., BWt=X*BWid, where 0<X<1). When the actual usage (BWu) falls below the threshold value (BWu<BWt), the LER sends a label request message with capacity equal to the actual aggregate bandwidth usage (BWu) to establish a new LSP.

BWid indicates that there is no other available LSP with less cost to accommodate the initial bandwidth demand. On the other hand, it is easy to infer from this affirmation that it is possible to find another LSP with equal or less cost that may satisfy a bandwidth demand smaller than BWid.

Consider that the network status of other links in the network that do not belong to this LSP remain unchanged. Based on this assumption, the probability of finding a new LSP from the same ingress node to egress node for BWid with less cost than the actually established LSP (LSP1) is equal to zero $P_{low\,cost}(BWid) = 0$. And then, the probability of finding an LSP with equal or greater cost is equal to 1. The probability of getting a new LSP for less cost with less bandwidth than the initial bandwidth demand, $P_{low\,cost}$(BW<BWid), increases when we decrease the BW demand with respect

to BWid. For this reason, it is important that the network manager be responsible for attempting the assignment of the appropriate value to BWt and waiting time based on the statistical data of the network. If the margin for triggering our mechanism is set too close to BWid (BWt has a value close to BWid), the LER triggers an LSP setup for slight changes of BWu with respect to BWid. As the probability to establish a new LSP with a high bandwidth demand is low, the probability of finalizing this procedure without success is high. In other words, high bandwidth requests have less probability of establishing a new LSP.

In the proposal the ingress LER not only monitors the decrement of LSP bandwidth usage, but also watches the released low-cost LSPs. When the traffic over the low-cost LSP ceases, and the associated LSP is released, the ingress LER must be capable of transferring traffic on the high cost LSP to the released low cost LSP. This improves network resource utilization and provides better overall performance for the MPLS based networks.

8.5 PROPOSED ALGORITHM

Figure 8.2 presents the flow diagram of the proposed algorithm. Though the flow diagram by itself is a formal description, we describe below our algorithm. It is important to explain the additional tables we include in the LERs. Apart from the normal Label Information Base forwarding table (LIB), we maintain two additional tables.

The first new table corresponds to the first rejected LSP on the optimal path for each LSP that was established using a non-optimal path. The data stored in this table are the LSPID, FEC, bandwidth and attempted output interface (link). Note that we put all LSPs whose optimal path is blocked and that are therefore currently using non-optimal LSPs in the rejected LSP table. Obviously, if the path is impossible to establish at any cost and the request is totally rejected, it implies no LSP was

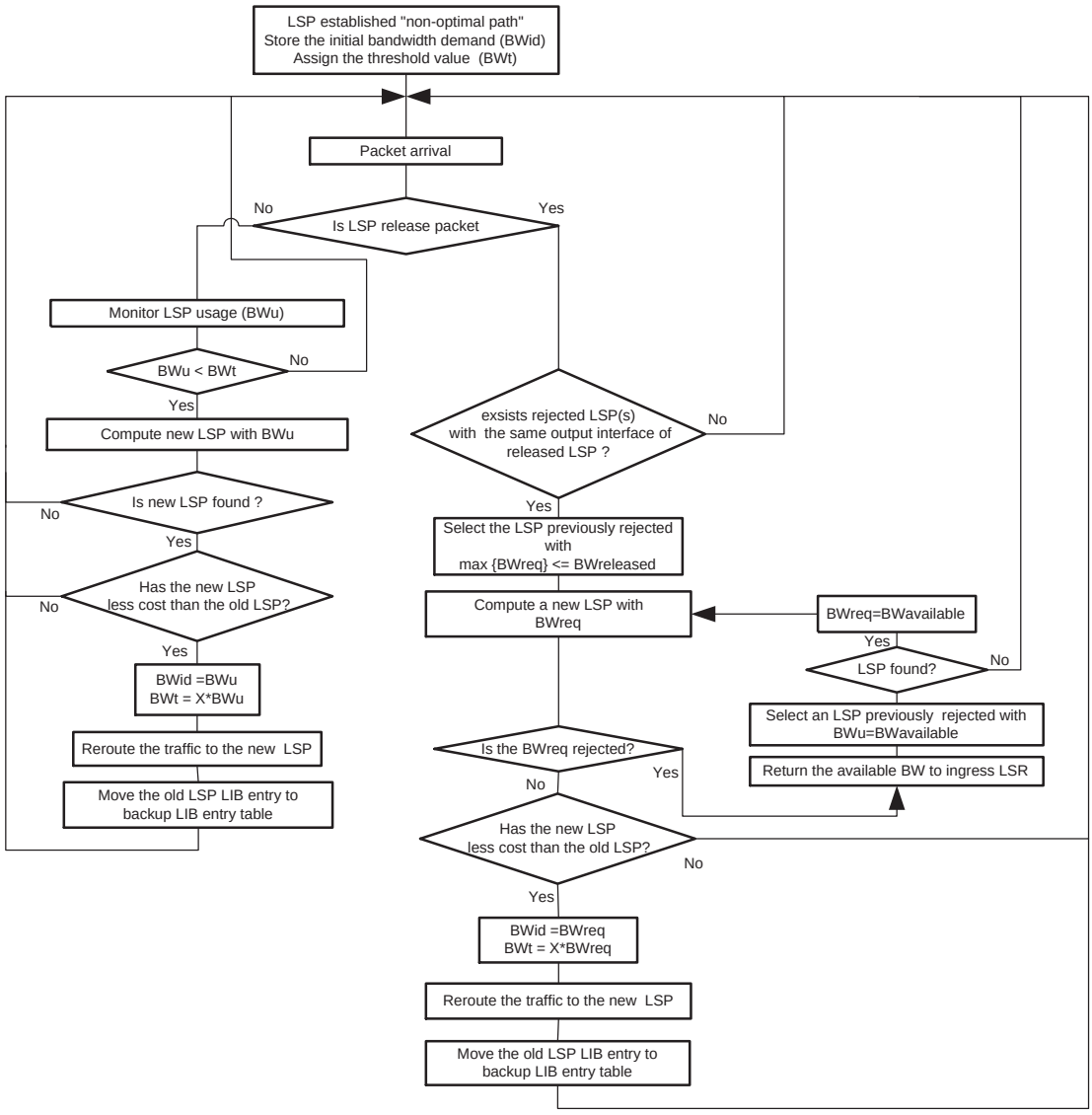

Figure 8.2 Flow diagram for proposed mechanism

established for this request: we ignore this and it is not included in the rejected LSP table.

The second table corresponds to a backup LSP information table of non-optimal rerouted LSPs, and we call this the "backup LIB entry table". In other words, it is the table entry formed by non-optimal LSPs removed from the LIB after traffic is

rerouted to an optimal LSP, with the only difference being that its bandwidth is set to zero (i.e., the same as reserving an alternative LSP without allocating reserved bandwidth). This backup LSP may be used for fast rerouting in case of failure (Chapter 3 and Chapter 6).

Our algorithm is mainly composed of two procedures: a bandwidth threshold procedure and a released LSP procedure.

8.5.1 Bandwidth threshold (BWt) procedure

The procedure starts when the LER receives any packet except for the LSP release packet. It then starts to monitor the LSP aggregate bandwidth usage (BWu). If the LSP bandwidth usage is less than the bandwidth threshold value (BWt) during a given period of time, the mechanism triggers the LSP compute procedure to compute a new LSP with the BWu request. If found, it compares whether the new LSP has a lower cost (shorter path) than the old one. If the result is yes, the algorithm updates BWid and BWt by BWu and X*BWu respectively for the newly established LSP, reroutes the traffic to the new LSP, and moves the old LSP LIB entry to the backup LIB entry table. If the new LSP has a higher or equal cost, it returns to the initial point in the process. It also returns to the initial point when it does not find a new LSP.

8.5.2 Released LSP procedure

All the necessary information is available in the LER. The procedure starts when the LER receives the LSP release message. After releasing the corresponding LSP, the algorithm looks up the rejected LSP table to verify if there are any rejected LSPs for this output interface. If the result is no, it returns to the initial point. If the result is yes, then the algorithm selects among all rejected LSP candidates (BW $\leq$ BWreleased) the one with maximum bandwidth. Note that this bandwidth corresponds to that being used on the current non-optimal LSP established for that request logged in the

rejected LSP table. Then, the mechanism starts to compute a new LSP with this BWreq. If this request is rejected due to a lack of available bandwidth in some of the segments of the new LSP that did not belong to the released LSP, the notification of reject message returns the amount of available bandwidth to the LER (ingress LSR). This information will be used to select the appropriate rejected LSP from the reject LSP table to establish a new LSP instead of using the bandwidth decremental algorithm. After that it seeks a reject LSP in the list that fulfills this condition. It then updates the BWreq with BW of the selected LSP from the rejected LSP table and starts the process of computing a new LSP. If it is not found, it returns to the initial point.

On the other hand, if the process is able to establish a new LSP, we compare this with the LSP we want to reroute (old LSP) in terms of the LSP length (cost). If the newly established LSP has a lower cost, we reroute the traffic, update the values of BWid=BWreq and BWt=X*BWreq for the newly established LSP, move the old LIB entry to the backup LIB entry table and return to the initial point. If the cost is equal or greater, the process returns to the initial point.

8.6 SUMMARY

Whereas the existing literature deals only with the problem of traffic demand, we have also focused on improving network resource allocation and utilization of MPLS networks in order to optimize the routing of IP traffic. We do this by: 1) dynamically adapting the LSP to the variations of the overall network load and 2) monitoring for released LSPs whose freed bandwidth can be allocated to a non-optimal LSP. We have shown that our enhanced mechanism allows for flexibility in network resource utilization, reduces delay by using the optimal available low cost path, and reduces new LSP request blocking.

Internet service providers generally must pay a fixed fee for the links they use to connect their routers. Obviously, they are interested in taking advantage of this fixed

cost by using optimal network resource utilization. As there is no mechanism for doing this automatically, the operator balances the load using certain criteria (rules) on a daily basis in response to measured link utilization.

Our proposal contributes also to better network resource planning. For example, normally the traffic volume in one direction is higher during the day than during the night. At off-peak hours our proposal plays an important role in rerouting the traffic from high cost paths to low cost paths. In fact, it is extremely likely there is an alternative path that would achieve better utilization and better overall performance. Rerouting traffic to a low-cost (or optimal) LSP reduces delay and delay variation and helps to improve the QoS for delay sensitive and multimedia applications.

The proposal reduces traffic blocking, and the delay that the traffic can experience traversing non-optimal paths. Besides better network utilization, our proposal would give truer figures of network resource utilization information to the network manager for network planning than that obtained by using non-optimal LSPs.

Finally, the specification of the threshold and the period of time to trigger the mechanism is an open issue in this proposal. The number of label request messages to set up a new LSP must be evaluated for different values of BWt and the timer. Moreover, though the proposal has good performance in simple network topologies, we think it needs to be proved in extended network topologies.