An Algorithm for Rerouting in Traffic Engineering of MPLS Based IP Networks

Muckai K Girish[*], Bei Zhou[†] and Jian-Qiang Hu[‡]

Abstract

Multiprotocol label switching (MPLS) enables IP networks with Quality of Service to be traffic engineered well. Rerouting and bumping of label switched paths (LSP) are caused by link or node failure or recovery, connection admission or load balancing. In this paper, we develop an algorithm for the traffic engineering problem associated with rerouting.

Keywords: Internet, Traffic engineering, IP networks, MPLS, QoS, Rerouting.

1 Introduction

Traffic engineering is an essential ingredient for guaranteeing QoS and for efficient or cost effective resource utilization, design and operation of IP networks. The MultiProtocol Label Switching (MPLS) working group at the IETF has been developing a standards-based approach for efficient IP packet transfer via traffic engineering. MPLS uses short, fixed-length, locally significant labels in the packet header and the packets are forwarded by network nodes via label swapping similar to layer 2 switching. The resulting *connections* are termed label switched paths (LSP) and a router that supports the MPLS protocols is called a label switching router (LSR). Traffic engineered paths can be established via MPLS signaling protocols such as CR-LDP ([6]) or RSVP ([3]), once the path is known. The determination of the optimal paths for the LSPs through the network requires the solution of an optimization problem such as the constraint based routing problem, which along with some requirements for traffic engineering were outlined in [1] and the optimization problems were mathematically formulated in [5]. This paper does not deal with the general traffic engineering problem and addresses the special case of rerouting of LSPs of different priorities.

---

[*] SBC Technology Resources, Inc., 4698 Willow Road, Pleasanton, CA 94588. Phone: (925) 598-1263. E-mail: mgirish@tri.sbc.com

[†] Dept. of Manufacturing Engineering, Boston University, 15 St. Mary's Street, Boston, MA 02215. Phone: (617) 353-4209. E-mail: bzhou@bu.edu

[‡] Dept. of Manufacturing Engineering, Boston University, 15 St. Mary's Street, Boston, MA 02215. Phone: (617) 353-9696. E-mail: hu@enga.bu.edu

In the next generation network architectures for the Internet backbone, MPLS is being considered for protection and restoration and fast rerouting functions as well which are traditionally done by the SONET layers as is evident from recent work in the IETF and other organizations. This motivates and drives more research into the area of the development of efficient algorithms and this paper falls into this realm.

Rerouting of LSPs in the MPLS layer can happen due to a number of reasons. When a new LSP is admitted into the network, if the available bandwidth on the route of this LSP is insufficient, then one or more lower priority LSPs may be rerouted over new paths which in turn can cause the bumping of even lower priority LSPs. When a link or a node (or some other component) fails, then the LSPs that are routed over this link or node need to be rerouted. Higher priority LSPs tend to get routed over more preferred paths which may cause lower priority LSPs to be bumped. Often, networks need to perform load balancing or re-optimization of the LSPs such that the network resources are optimally allocated. For example, this can happen when a link or a node recovers from a failure, a new path is available and so some of the LSPs can be rerouted over these less or non-congested links. This situation also arises when a new network element such as a link or a node is configured. One of the requirements for effective management of IP networks is to have intelligent methods to deal with rerouting and bumping. In this paper, we mathematically formulate the rerouting phenomena and develop an algorithm for this in order to aid in the expeditious development of the algorithms for traffic engineering.

2 The Rerouting Problems

Let $G = (V, E, C)$ be a graph describing the physical topology of the network, where V is the set of nodes, E is the set of links (which are defined as directed arcs) and C is the set of capacity and other constraints associated with the nodes and links. Let $H = (U, F, D)$ be the induced MPLS graph where U is the set of LSRs where one or more LSPs originate or terminate, F is the set of LSPs and D is the set of demands associated them. Let $u_l, v_l, \mu_l, a_l, K_l$ be the originating LSR, terminating LSR, available bandwidth, administrative cost and oversubscription factor, respectively, of link l. We assume that a total of L links are in the network. Let $\lambda_j, s_j, d_j, h_j, \alpha_j, \beta_j$ be the effective or equivalent bandwidth, ingress LSR, egress LSR, maximum allowed number of LSR hops through the network, set-up priority, holding priority, respectively, of LSP j. We assume that $0 \leq \alpha_j, \beta_j \leq \alpha^*$ and the convention that we employ in this paper is that for α_j and β_j, a lower numerical value indicates higher priority. Moreover, we assume that for all LSPs, $\alpha_j \geq \beta_j$ since we do not want an LSP with certain set-up and holding priorities to bump a previously established LSP with identical priorities. Let x_{jl} be the binary decision variable which equals 1 if LSP j is routed on link l, and equals 0 otherwise.

2.1 The Algorithm

Let G be the network topology graph, after the state change that triggered the rerouting. Let H be the induced demand graph of the LSPs that are currently routed in the network which are not affected directly by the state change and let F^r be the set of LSPs that are to be re-routed (in subsequent sections, we show how to derive F^r in a variety of situations). We propose an algorithm below for rerouting and bumping. The inputs to this algorithm are $G, F^r, x^*_{jl}, j \in F$ and the outputs are the explicit routes ($x^*_{jl}\ \forall\ j$) of the LSPs through the network.

- Divide the set of LSPs to be re-routed, F^r into subsets F^r_k according to set-up priority k and order the subsets, where

  $$F^r_k = \{j \in F^r | \alpha_j = k\}\ \forall\ k = 0, 1, \ldots, \alpha^*$$

- Start with the highest set-up priority LSPs; i.e. set $k = 0$

- **Step (A)** If the set of LSPs of priority k is empty, i.e. $F^r_k = \phi$, go to the next lower set-up priority of LSPs (increase k by 1) and Go back to Step (A); otherwise:

  - Split the set of LSPs, F into F_h and F_l which constitute the set of LSPs that are of higher or lower holding priorities than k, respectively:

    $$\begin{aligned} F_h &= \{j \in F | \beta_j \leq k\} \\ F_l &= \{j \in F | \beta_j > k\} \end{aligned}$$

  - As LSPs with set-up priority k cannot bump LSPs of higher holding priorities F_h and hence they are not rerouted, pin or fix the routes for F_h, where

    $$\begin{aligned} x^h_{jl} &= x^*_{jl}\ \forall\ j \in F_h\ \forall\ l \in E \\ \mu^h_l &= \mu_l - \sum_{j \in F_h} \lambda_j x^h_{jl}\ \ \forall\ l \in E \end{aligned}$$

  - Find the optimal solution (or the best set of feasible solutions), x^*_{jl} of the following constraint based routing problem with G as the network topology graph with pinned LSPs F_h and with LSP demand F^r_k:

    $$\text{Minimize} \quad Z = \sum_{l \in E} \sum_{j \in F^r_k} a_l \lambda_j x_{jl} \tag{1}$$

subject to:

$$\begin{aligned}
\sum_{j \in F_k^r} \lambda_j x_{jl} &\leq \mu_l^h K_l, \;\; \forall\, l \in E \\
\sum_{l \in E} x_{jl} &\leq h_j, \;\; \forall\, j \in F_k^r \\
\sum_{\forall l | u_l = n} x_{jl} &= 1 \;\; \forall\, n \in U \;\; \forall\, j | s_j = n \\
\sum_{\forall l | v_l = n} x_{jl} &= 1 \;\; \forall\, n \in U \;\; \forall\, j | d_j = n \\
\sum_{\forall l | u_l = n} x_{jl} - \sum_{\forall l | v_l = n} x_{jl} &= 0, \;\; \forall\, n \in V \;\; \forall\, j | s_j \neq n \text{ or } d_j \neq n \\
0 \;\leq\; x_{jl} &\leq 1, \;\; \forall\, j \in F_k^r, l \in E \text{ and integer}
\end{aligned}$$

- ∗ In the above optimization problem, the objective function minimizes the sum over all links of the product of the administrative cost and the total flow in each link. The constraints represent the following scenarios. The first constraint ensures that the link capacities are not exceeded. The second constraint restricts the number of hops in the path of a LSP. The next three constraints assure that all LSPs originating and terminating, respectively, in a node are routed and also ensure that the LSPs are routed through intermediate nodes, thereby, ensuring an end-to-end path through the network. The last constraint specifies that all decision variables are either 0 or 1.
- ∗ If an optimal solution to the above problem exists, then new routes for the lower priority LSPs, $x^*_{jl}, j \in F_k^r$ are obtained. We add these newly obtained LSP routes to the set of optimally routed LSPs. In other words, set $F^* = F_h \cup F_k^r$. Go to Step (B)
- ∗ If the constraints do not lead to a feasible solution to the above problem, then we assume that we can find the best possible feasible solution $x^*_{jl}, j \in \overline{F} \subset F_k^r$ for the set of LSPs $\overline{F}$ that have new routes; The other LSPs $(F_k^r - \overline{F})$ cannot be re-routed. We now add the feasible LSPs to the set of optimally routed LSPs; i.e. set $F^* = F_h \cup \overline{F}$. Go to Step (B)

– **Step (B)** In order to check if the newly obtained optimal routes for LSPs along with any remaining lower priority LSPs can still be routed feasibly, we add F_l to F^* and check for any violation of the link capacity constraints:

$$F_{lv} = \{ j \in F_l | \sum_{j \in F^* \cup F_l} \lambda_j x_{jl} > \mu_l K_l, \;\; \forall\, l \in E \}$$

- If all the capacity constraints are satisfied (i.e. $F_{lv} = \phi$), then the set of LSPs $F^* = F^* \cup F_l$ have new routes. We then remove these from the set of LSPs to be rerouted; i.e. $F^r = F^r - F^r_k$. Go to the next lower set-up priority of LSPs (increase k by 1) and go to Step(A)
- If one or more of the constraints are violated (i.e. $F_{lv} \neq \phi$), then isolate the LSPs that violate the link capacity constraints, F_{lv}. $F^* = F^* + \{j \in F_l | \text{feasible solution}\}$. $F^r = F^r - F^r_k$. Go to the next lower set-up priority of LSPs (increase k by 1) and add these LSPs that cause capacity violation to the set of LSPs that need to be rerouted; i.e. set $F^r = F^r \cup F_{lv}$. Now order F^r into $F^r_k, F^r_{k+1}, \ldots, F^r_{\alpha^*}$, where $F^r_p = \{j \in F^r | \alpha_j = p\}$. Go to Step (A)

The algorithm presented above is based on a centralized implementation model. It is, however, possible to apply this in a distributed environment in which each LSR makes its own decisions on rerouting LSPs that originate in it. This can be done as follows. The routing protocol (for example, OSPF [7] and IS-IS, [2]) with extensions for traffic engineering floods the network with link state information which also contains additional metrics such as utilization, delay and color of links in the network (see [4]). Each node then constructs its own view of the network topology from this information. For each LSP to be routed on the network the shortest path is computed by the originating node of the LSP (hence called source routing). First, it prunes the network topology such that only those links and nodes are included in this topology that satisfy the requirements of the LSP such as the link color and bandwidth. It then computes the shortest path through this pruned topology graph and the LSP is routed along this path. Therefore, this approach is a special case of the algorithm considered in this paper as we can make our algorithm distributed by considering one LSP at a time.

2.2 Connection Admission Control

When a request for a LSP set up comes in, the connection admission control problem is solved to decide whether to reject or to admit this LSP. If the LSP can be admitted, then the solution also determines the route of this LSP through the network. In addition, a decision has to be made whether to bump (reroute) one or more existing LSPs or not. If bumped, then alternate paths have to be found for these bumped LSPs, if possible. Moreover, these may bump additional LSPs along the new route based on the affinity of the routes and the set up and holding priorities of the LSPs. The rerouting/bumping algorithm can be applied to solve this problem. Let c be the LSP that was admitted with optimal route x_{cl}. Then the set of LSPs that have to be rerouted can be determined from:

$$F^r = \{j \in F \text{ with Policy } P | (\lambda_c x_{cl} + \sum_{j \in F} \lambda_j x_{jl}) > \mu_l K_l, \ \ \forall\, l \in E \text{ with } x_{cl} = 1\}$$

Note that a certain policy P is applied for selecting the LSPs; some examples of which could be to first select the LSPs with the lowest holding priorities and breaking ties based on lower effective bandwidth and lower set-up priorities etc.

2.3 Link/Node Failure and Recovery

When link g fails, the topology graph changes by $E = E^* - \{g\}$ and the set of LSPs to be rerouted becomes $F^r = \{j | x_{jg} = 1\}$. In case of a node failure, all links originating and terminating on that node are down leading to possible rerouting the LSPs that are routed over those links. The rerouting/bumping algorithm developed in this paper can then be applied to these failed links. Note that the LSPs that originate or terminate on the failed node cannot be routed and so are torn down.

When link g recovers from a failure, then the topology graph changes by $E = E^* + \{g\}$. When a node recovers from a failure, the LSPs that originate or terminate in that node that were torn down when failure happened, now can be re-established. The traffic engineering problem for routing these LSPs is similar to the connection admission control problem. If more than one LSP is involved, then the order in which they are rerouted is based on the policy that the network (or in some cases restricted by the LSR) implements. For all other LSPs that were rerouted or torn down when the node failed or in the case of a link recovery, the load balancing or periodic optimization methods can be used.

2.4 Load Balancing/Periodic Optimization

Load balancing or periodic optimization is often needed for optimal network utilization. Some example scenarios are: a link or a node comes up (either from failure or the provisioning of a new link or node), one or more LSPs are torn down causing imbalance on the trunk utilization. The types of measures that are used to determine the imbalance on the load on the trunks (changes by a certain amount or proportion) could be several, including the configured utilization (based on the equivalent bandwidths of the LSPs) on a link or the real-time utilization of a link (could be instantaneous or time average or some other function). Let ρ_l be the utilization of link l. Let b be the allowed deviation/spread in utilization among the links. Load balancing may be done on the LSPs on link l when the following criteria is violated:

$$\overline{\rho} - b \leq \rho_l \leq \overline{\rho} + b \tag{2}$$

where $\overline{\rho}$ is the average utilization on all links. One can identify a LSP at a time that is routed over a trunk. For example, an LSP with the lowest set-up priority, α_j, is first chosen so that the probability of bumping taking place are minimized. One can apply the connection admission control algorithm for this LSP. Note that in this case, the objective function can be modified with a load balancing criteria. The solution to this problem gives the new path for this LSP which can be set up followed by the tearing down of the old path. Then, the utilization on the links can be compared again to decide whether more reroutes are needed for optimal load balancing.

The frequency and the number of LSPs rerouted during load balancing should be minimized, thereby avoiding or reducing route flaps and oscillations, minimizing the amount of processor horsepower consumed for executing the traffic engineering problem and tearing down and setting up of LSPs, and minimizing the impact on data traffic flow (such as packet loss and packet reordering) due to rerouting. Bumping of other LSPs should be prevented for load balancing purposes since one does not want to cause a chain reaction of LSP reroutes.

2.5 Numerical Examples

The underlying optimization problem that need to be solved is, in the general case, an NP-complete problem and hence questions such as the computational complexity and burden in solving this problem arises. Nevertheless, this problem has been studied in detail and many solutions have been proposed many of which are heuristics which work well under certain conditions. We have been working on the development of efficient solutions and the method based on Lagrangian relaxation (see [9]) and detailed numerical results will be published in a forthcoming paper. Here, we provide two examples to illustrate that our method indeed works well.

In the first example, we consider a network with 9 nodes and 16 links with 66 LSPs. When formulated as an Integer programming problem, we have 2673 binary decision variables. Using CPLEX to solve this problem, the optimal cost of the objective function works out to $f^* = 12590$. When the Lagrangian relaxation algorithm was applied to this problem, the results are shown in Figure 1. Note that our algorithm produces lower and upper bounds and it can be seen that the solution is very close to the optimal value in less than 20 iterations itself.

In the second example, the network has 29 nodes, 61 links and 140 LSPs. In this case, the number of binary decision variables works out to 58870. CPLEX provided the optimal cost to be $f^* = 9010$. Figure 2 shows the results of our algorithm based on Lagrangian relaxation. Once again, this leads to convergence and the method leads to a solution very close to optimality in just over 50 iterations. The computation time for both the examples are negligible.

3 Conclusions

The success of next generation IP networking depends on the ability to offer and support QoS to customers. It is clear that traffic engineering is critical for this as well as for efficient network resource utilization and operation. In this paper, we formulated and developed an algorithm for characterizing the rerouting and bumping phenomena in MPLS networks. The algorithm can be used to determine the rerouted paths of LSPs. We then showed how the rerouting/bumping can be applied to various problem scenarios such as connection admission, link or node failure or recovery and load balancing. Our on-going research deals with the development of efficient solution methods for the traffic engineering problems which, unfortunately, fall into the realm of NP-complete optimization problems ([8]).

References

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

[2] H. Smit and T. Li, "IS-IS Extensions for Traffic Engineering," Work in Progress, *Internet Draft draft-ietf-isis-traffic-01.txt,* May 1999.

[3] D. Awduche, L. Berger, D. Gan, T. Li, G. Swallow and V. Srinivasan, "Extensions to RSVP for LSP Tunnels," Work in Progress, *Internet Draft draft-ietf-mpls-rsvp-lsp-tunnel-03.txt,* September 1999.

[4] D. Fedyk and A. Ghanwani, "Metrics and Resource Classes for Traffic Engineering," Work in Progress, *Internet Draft draft-fedyk-mpls-te-metrics-00.txt,* October 1999.

[5] M. K. Girish, B. Zhou and J. Q. Hu, "Formulation of the Traffic Engineering Problems in MPLS based IP Networks," *The Fifth IEEE Symposium on Computers and Communications,* Antibes- Juan les Pins, France, 2000 (to appear).

[6] B. Jamoussi et al., "Constraint-Based LSP Setup using LDP," Work In Progress, *Internet Draft draft-ietf-mpls-cr-ldp-03.txt,* September 1999.

[7] D. Katz and D. Yeung, "Traffic engineering extensions to OSPF," Work in Progress, *Internet Draft, draft-katz-yeung-ospf-traffic-00.txt,* April 1999.

[8] C. H. Papadimitriou and K. Steiglitz, "Combinatorial Optimization: Algorithms and Complexity," *Prentice Hall, Inc.,* 1982.

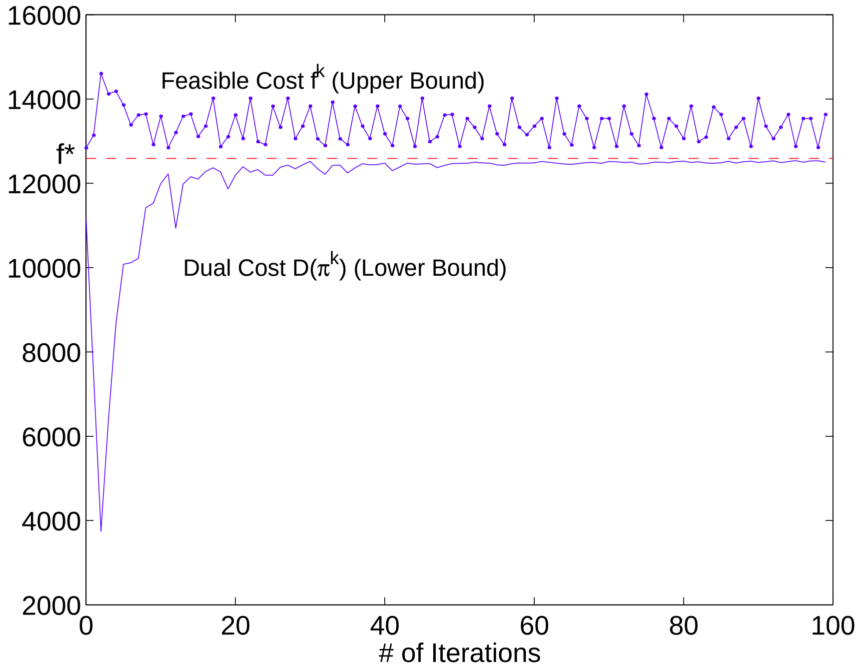

Figure 1: Dual and Feasible Cost in the first example

[9] B. Zhou, "Constraint Based Routing Problem in MPLS Based IP Networks," *M. S. Thesis,* Dept. of Manufacturing Engineering, Boston University, 2000.

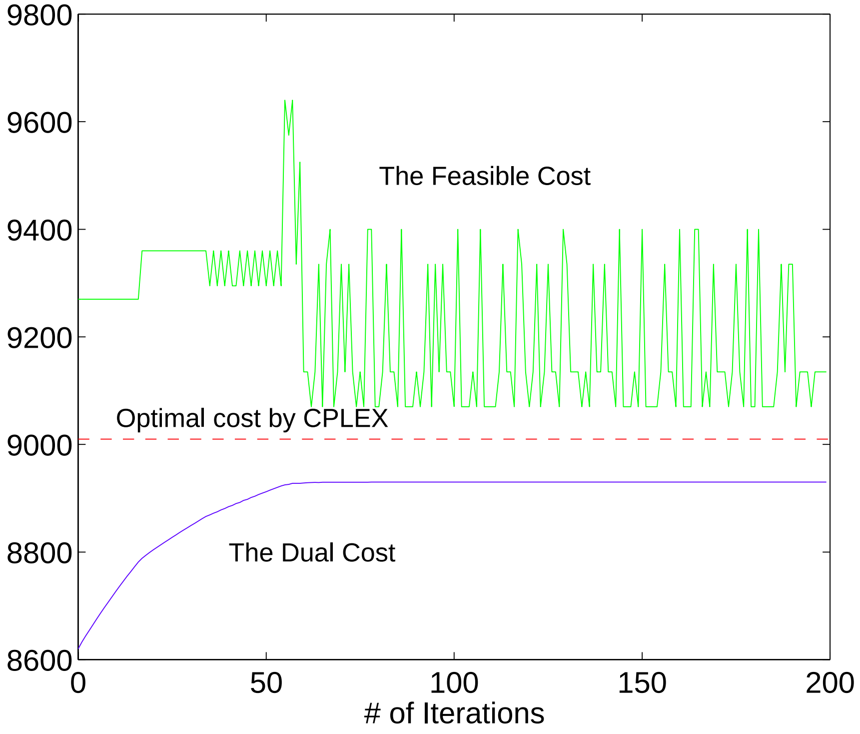

Figure 2: Dual and Feasible Cost in the second example