Strategy-Based Path Selection for Traffic Engineering over MPLS

Lei Li*, Yangchun Li**, Hitoshi Yamada***
*Fujitsu Research & Development Center
Beijing, China
lilei@frdc.fujitsu.com
** China Telecom-Guangzhou
Research & Development Center
Guangzhou, China
liyc@gsta.com
***Fujitsu Laboratories Limited
Kawasaki, Japan
hitoshi@labs.fujitsu.co.jp

Abstract — **The advantage of Multi-protocol Label Switching (MPLS) is its capability to route the packets through explicit paths. And for traffic engineering over MPLS, the key is to select suitable paths to balance the network load to optimize network resource utilization and traffic performance. And from network carriers' point of view, such optimization should be based on their individual networks, service requirements as well as their own administrative policies. In this paper, we present a method to provide the capability to control traffic engineering so that the carriers can define their own strategies for optimizations and apply them to path selection for both QoS routing and dynamic load balancing.**

Keywords-MPLS; traffic engineering; optimizations; control strategies; QoS routing; dynamic load balancing

I. INTRODUCTION

With explosive growth of Internet users and new bandwidth-consuming services, many network carriers are facing the problem of how to accommodate such ever-growing demands for bandwidth. And the static nature of current routing algorithms, such as OSPF or IS-IS, has made the situation worse since the traffic is concentrated on the "least-cost" paths which causes the congestion for some links while leaving other links lightly loaded. Therefore, MPLS traffic engineering is proposed and by taking advantage of MPLS, traffic engineering can route the packets through explicit paths to optimize network resource utilization and traffic performance as stated in [1]. And the calculation of these explicit paths, or path selection, plays the important role in traffic engineering since it decides the traffic distribution in the network.

For such path selection, there are two major criterions, one is the link loads of the path and the other is its hop count. With the increase of link loads, the packets on this path will suffer more queuing delay; while the increase of hop count will produce more propagation and processing delay. Intuitively, there should be some kind of trade-off between these two criterions if we can not optimize both. And based on biasing in such trade-off, the current path selection methods can be classified into following three categories

1. Hop-unlimited: such approaches map the traffic to the least loaded paths through Least Load Routing or Bernoulli Splitting as in [2] and [3];
2. Minimal-hop-only: such approaches select the least loaded paths only within the minimal hop paths as in [4];
3. Shortest-distance: such approaches define the link costs based on the loads, or available bandwidths, of these links and the path with minimal total cost is selected;

There are a lot of the comparative studies for the approaches above. And in [5] and [6], the authors present a systematic evaluation for these algorithms, and the results show that, for traffic with bandwidth guarantees, a routing algorithm that gives preference to limiting hop count performs better when the network load is heavy, while an algorithm that gives preference to less link loads (preferring load balancing) performs better when the network load is light. On the other hand, for best-effort traffic, the shortest-distance algorithm has a clear performance edge over other algorithms. These results show that no routing algorithm is always optimal. The carriers have to make the decision based on their traffic distributions and the service requirements. This is often difficult for carriers, if not impossible, because of the unevenly traffic distribution as well as the different service requirements. And if the condition changes, they have to totally change their routing algorithms. Furthermore, based on the nature of the network under control, the strategy for the tradeoff between limiting hop count and preferring load balancing should also be different. Just for an example, in a metro-area network, the physical span of the network is relatively small, increase of hop counts will not bring much extra propagation delay, we can give more preference to load balancing to maximize the total throughput; while in country-wide backbone networks, the case is totally different because one extra hop may mean more than one thousand kilometers. Therefore, for different networks, network carriers may have different administrative policies for

0-7803-8836-4/04/$20.00 ©2004 IEEE.

traffic engineering. Unfortunately, currently no published routing algorithm can incorporate such administrative policies.

In this paper, we present a simple method to enable the carriers to deploy their own strategies to select the suitable paths for load balancing. This method is based on the shortest-distance algorithm. And just through adjusting the value of one scaling parameter for link loads, the tradeoff between limiting hop counts and preferring load balancing is simply manipulated. And we also present that how the administrative policies can be incorporated in path selection in our method.

II. UTILIZATION-BASED SHORTEST-DISTANCE ALGORITHMS

In utilization-based shortest-distance algorithms, the link cost is defined to increase in a convex manner with the increasing link load to give penalty to the links with large utilizations. And the path with the minimal total cost is selected. And the definition for the cost function is either exponential as in [4], [7] and [8] or reciprocal as in [9] and [10].

The exponential approaches use the link utilizations either as the base as in [7] or as the index as [4] and [8]. And the advantage of these approaches is that the shape of the convex functions can be controlled by choosing the value for the other parameter in the exponent function, either index or base, so that the limited capability for manipulating the trade-off between limiting hop count and preferring load balancing is provided. For an example, in [7], the authors propose to update the base of link utilizations in link cost function when these utilizations cross the predefined threshold so that limiting hop count is preferred when the utilization is high and load balancing is encouraged when the utilization is light. Obviously, the drawback for these exponential approaches is their complexity in control of the shapes of the exponential functions based on the network status.

On the contrary, in the reciprocal approaches, the link cost is just defined as either the link capacity or the current load divided by the residual bandwidth, that is, $\frac{B}{B-L}$ as shown in [9] or $\frac{L}{B-L}$ as shown in [10], where B and L is the capacity and the load of the link, respectively. For simplification, the equations above are transformed to $\frac{1}{1-U}$ (1) and $\frac{U}{1-U}$ (2), while U is the utilization of the link. Based on equations (1) and (2) it is very simple to calculate the link cost. However, this time we totally lose control on the trade-off between limiting hop counts and preferring load balancing as shown below.

The following simple example is to illustrate why we can not control load balancing if the shortest-distance routing algorithm is used for path selection.

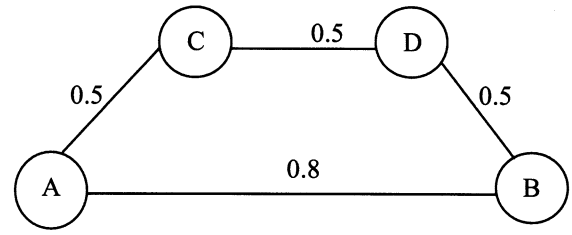

Figure 1. Simple Example for Cost Function $1/(1-U)$

In this simple example, between the ingress node A and egress node B, there are two available paths, the utilization for each link is denoted above itself in the figure. If equation (1) is used to calculate the link cost, the total cost for minimal hop path (path A-B) is 5 and the total cost for the other path (path A-C-D-B) is 6. In this case, the path A-C-D-B will not be taken as the alternative path for load balancing since its cost is still larger than that of minimal-hop path. More general analysis is given below. Assume for one ingress-egress pair, the current path consists of n ($n \geq 1$) links with utilizations as u_i (i=1… n). At the same time, there is another path for this pair, which consists of $n+m$ ($m \geq 1$) links with utilizations as u_j (j=1…$n+m$). And the total cost for these two paths is denoted as C_{cur} and C_{alt}, respectively. In order that this longer path is selected as alternative path for load balancing, the condition $C_{alt} \leq C_{cur}$ (3) should be satisfied. Based on equation (1), the condition (3) can be expressed as below

$$\sum_{j=1}^{n+m} \frac{1}{1-u_j} \leq \sum_{i=1}^{n} \frac{1}{1-u_i} \qquad (4)$$

Define $U_{alt} = 1 - \frac{1}{C_{avg\_alt}}$ and $U_{cur} = 1 - \frac{1}{C_{avg\_cur}}$ while C_{avg_cur} and C_{avg_alt} are the average costs for links in current and alternative path respectively. In this sense, the U_{alt} and U_{cur} are the "average" link utilizations according to the link cost, then inequality above can be expressed as

$$\frac{n+m}{1-U_{alt}} \leq \frac{n}{1-U_{cur}} \qquad (5)$$

Therefore, we have the condition for the link utilizations to select the alternative path, that is

$$U_{cur} \geq \frac{m+n\times U_{alt}}{m+n} \qquad (6)$$

From equality (6), we can find that if the equation (1) is used as link cost function, whether load can be balanced to alternative paths is based on the number of hops in current and alternative paths as well as the link utilizations in these paths. As far as the practical operations are concerned, this means that the carriers can not control whether and when to balance their network load. Through the similar way, we can get the condition for load balancing when link cost function (2) is used, that is

$$U_{cur} \geq \frac{m+n}{m+n\times U_{alt}} \times U_{alt} \qquad (7)$$

From (6) and (7), we can see that the shortest-distance routing based on reciprocal cost functions incorporate the tradeoffs between limiting hop counts and preferring load balancing. However, such tradeoffs are static. Obviously, it is undesirable for network carriers to control traffic engineering with such static tradeoffs as shown in (6) and (7) in different networks with ever changing traffic.

In the next section, we present our method which not only compromises the simplicity of reciprocal approaches to calculate link costs and the capability of exponential approaches to control the shapes of the cost functions but also facilitates the carriers to control the traffic engineering according to their own administrative policies.

III. STRATEGY-BASED PATH SELECTION

The key concept in our proposed Strategy-based Path Selection is to scale the link utilizations by a factor called *Util_weight* and the value of *Util_weight* is chosen based on the strategies to control traffic engineering. Applying this scaling factor to link utilizations in equation (1), we have

$$C_{link} = \frac{1}{1 - Util\_weight \times U} \quad (Util\_weight \geq 0) \tag{8}$$

Through the similar way to conclude (6) in the previous section, we have the condition to select alternative paths in this modified link cost function as follows

$$Util\_weight \times U_{cur} \geq \frac{m + n \times Util\_weight \times U_{alt}}{m+n} \tag{9}$$

Therefore, we have the following inequality

$$U_{cur} \geq \frac{\frac{m}{Util\_weight} + n \times U_{alt}}{m+n} \tag{10}$$

Comparing equality (10) with (6), we can see now the path selection is influenced by the scaling factor, *Util_weight*. And how to choose the value for *Util_weight* depends on the different schemes of traffic engineering as well as control strategies, which will be shown in the later sections. In short, for the complete same current path and alternative path, when *Util_weight*<1, the threshold on link utilizations to take the alternative path is increased, this means that the limiting hop count is preferred. On the other hand, when *Util_weight*>1, this threshold is decreased to prefer load balancing to longer path.

A. *Util_weight for QoS Routing*

Quality-of-Service routing often use the nonminimal hop paths to route the traffic with bandwidth guarantees if the minimal hop paths do not have the enough bandwidth. However, previous comparative studies have shown that routing traffic to longer routes consumes additional network resources at the expense of future connections, which may be unable to locate a route. Therefore, as stated in [15], it is necessary to consider the total resource allocation for a flow along a path, in relation to available resources, to determine whether or not the flow should be routed on the path. The goal of such "high level admission control" mechanism is to ensure that the "cost" incurred by the network in routing a flow with a given QoS is never more than the revenue gained considering the revenue that may be the lost in potentially blocking other flows that contend for the same resources. In this section, we will show that our proposed Strategy-based Path Selection can implement such mechanism through using *Util_weight* to set the eligibility of the paths for QoS routing based on the network status and control strategies.

In our proposed method, we first find the maximal link utilization for each available path and the utilization for links in a path is defined as the maximal link utilization in this path. Assuming that there are n hops in the minimal-hop path for an ingress-egress pair, and then based on equation (8), the maximal total cost for this path is

$$C_{\min\_hop} = \frac{n}{1 - Util\_weight} = n + \frac{Util\_weight}{1 - Util\_weight} \times n \tag{11}$$

Assuming that for this ingress-egress pair there is another available path with $n+m$ ($m \geq 0$) hops, the cost for this path is minimal when there is no load on this path and this minimal cost is $(m+n)$. In shortest-distance routing, this path will be selected for routing traffic only if its cost is equal or less than that of the minimal-hop path, which is shown in (11). Therefore, we get the upper bound of the number of extra hops of this path if it is eligible for routing traffic between this ingress-egress pair, that is,

$$m \leq \frac{Util\_weight}{1 - Util\_weight} \times n \tag{12}$$

And if this path is not empty and its maximal link utilization is U, this path will be selected only if

$$\frac{m+n}{1 - Util\_weight \times U} \leq \frac{n}{1 - Util\_weight}$$

Hence, we have the upper bound for maximal link utilizations for this path, that is,

$$U \leq 1 - \frac{1 - Util\_weight}{n \times Util\_weight} \times m \tag{13}$$

Through inequality (12), we set the range of the paths eligible for QoS routing based on their lengths. And through (13), the upper bound of maximal link utilization is specified for an eligible path. And such bounds are both definable based on the control strategies through adjusting the value of *Util_weight*.

In this paper, we set the value of *Util_weight* as follows

$$Util\_weight = \frac{m}{(m+n) - n \times U} \tag{14}$$

while n is the number of hops in minimal-hop path and $m \geq 0$, $0 \leq U \leq 1$. With this *Util_weight*, a nonminimal path whose hop count is $(n+m)$ and maximal link utilization is U will have the same cost as the maximal cost of the minimal-hop path with n hops. We call this nonminimal hop path as the "worst" path,

which sets the bottom line for the paths eligible for QoS routing. Assuming that for the ingress-egress pair as shown in the analysis above, there is one available path, called path A, with $(n+h)$ $(h \geq 0)$ hops and maximal link utilization as Z. With the *Util_weight* as shown in equation (14), the cost for path A is

$$\frac{n+h}{1-\frac{m \times Z}{m+n-n \times U}} = \frac{n+h}{n+m} \times \frac{1}{1-\frac{n \times u + m \times Z}{n+m}}$$

If path A is eligible for QoS routing, its cost must be equal or less than the maximal cost of the minimal-hop path, that is

$$\frac{n+h}{n+m} \times \frac{1}{1-\frac{n \times u + m \times Z}{n+m}} \leq \frac{n+m-n \times U}{1-U} \quad (15)$$

If path A is longer than the "worst" path, that is, $h>m$, from (15) we have

$$\frac{1}{1-\frac{n \times u + m \times Z}{n+m}} < \frac{n+m-n \times U}{1-U}$$

Therefore

$$\frac{n \times U + m \times Z}{n+m} < U \quad (16)$$

Hence, we get the upper bound for maximal link utilization in path A, that is $Z<U$. And this shows that if a path is longer than the "worst" path, it will be selected for QoS routing only if its maximal link utilization is less than that of the "worst" path, which is specified by U. In short, with given U and m, the range for the length of eligible paths is set to $[n, \frac{m}{1-U}]$. And the paths which are longer than $n+m$, will be taken only if their maximal link utilization is less than U. Through this way, the location of network resources to nonminimal hop paths is controlled dynamically according to the network status and control strategies. The following is an example for how to use our method to control QoS routing.

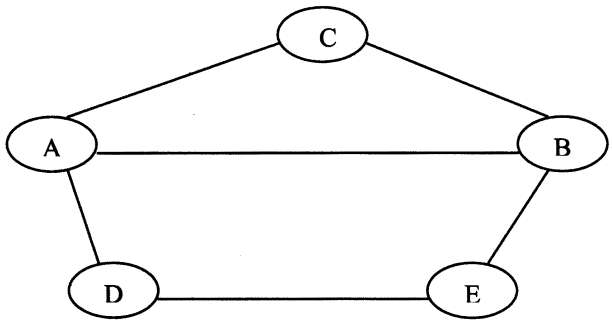

Figure 2. Example Network for Strategy-based Path Selection

The network topology is shown in the figure 2 and the interested ingress-egress pair is node A and node B. By default, all traffic for this pair is forwarded through the minimal-hop path. Based on the nature of this network and the traffic distributions, we can define the control strategy like follows "the nonminimal hop path with more than one extra hop should not be taken unless its maximal link utilization is less than 80%". For current QoS routing algorithm, it is impossible to incorporate such control strategies. But in our method, we just calculate the value for *Util_weight* based on equation (14) with n=1, m=1 and U=0.8. With this *Util_weight*, Path A-C-B may be used to route traffic for this pair if its maximal link utilization is less than 80%. And if not, path A-D-E-B may be selected if its maximal link utilization is below 60%; otherwise, the incoming flows will be rejected if there are not enough resources for them on minimal-hop path. As we can see from this example, in our method, the hop counts of nonminimal hop paths are limited based on their own maximal link utilizations so that the bandwidth is saved for future connections. In this example, at least 20% bandwidth is saved for the future connections between node A and node C as well as those between node C and node B. And for the same network, if the control strategy is changed to give more preference to these future connections, we just use a smaller U to recalculate the *Util_weight* and assign it to this ingress-egress pair.

In short, by use of our proposed Strategy-based Path Selection in QoS routing, we set the bottom line for selecting the nonminimal hop paths for QoS routing based on their number of hops and maximal link utilizations. Longer paths (specified by m) will be selected only if there are lightly loaded (specified by U). Through this way, the location of the network resources for nonminimal hop paths is controlled dynamically based on the network status and control strategies.

With the flexibility and simple computation, the value of *Util_weight* can be same for the whole network, with n as the average hop count for minimal hop paths; or different for individual ingress-egress pairs to deploy different control strategies. Furthermore, even for single ingress-egress pair, different *Util_weight* can be assigned to traffic with different priorities to support multi-class QoS routing.

B. Util_weight for Dynamic Load Balancing

Utilization-based shortest-distance routing is also often used in Dynamic Load Balancing systems such as those proposed in [11] and [12], whose objective is to maximize the network throughput for best-effort traffic. In dynamic load balancing, there is a user-defined high threshold for link utilizations. And once the utilizations of some links are above this high threshold, the paths over these links are considered as congested. Then alternative paths should be selected for these congested paths and traffic should be rebalanced between the congested paths and alternative paths to reduce the utilizations of the congested links below the high threshold. Therefore, in path selection for dynamic load balancing, the key is that the alternative path should never take the links whose utilizations are already above the high threshold. Otherwise, the congestion can not be solved. However, with current shortest-distance routing algorithms, we can not avoid such cases because current routing algorithms do not take the user-specified high threshold into consideration. Still taking the network shown in figure 1 as the example, this time we use the dynamic load balancing in this network with equation (1) as link cost function and the high threshold is set to 70% to force

load balancing. But such control will be tried in vain due to the static nature of equation (1) in tradeoff between limiting hop count and preferring load balancing. According to equality (6), for this example, the alternative path will still take the already-congested link unless the "average" link utilization on minimal-hop path is above 84%.

Such problems can also be resolved through using *Util_weight* in the calculations for link cost and the corresponding link cost function with *Util_weight* is as follows

$$C_{link} = \begin{cases} \dfrac{1}{1 - Util\_weigth \times U}, & (Util\_weight \times U) < 1 \\ Max\_Cost, & (Util\_weight \times U) \geq 1 \end{cases} \quad (17)$$

The difference between (17) and (8) is that now the costs of the links whose utilizations are higher than $\frac{1}{Util\_weight}$ are assigned with a very large value, called *Max_Cost*. As far as the link utilizations are considered, this cost function puts a high threshold for them, which is $\frac{1}{Util\_weight}$. Once the utilizations of some links reach this threshold, the alternative path will not take these links because now their costs are very large in shortest-distance algorithm. By setting $Util\_weight = \frac{1}{High\_threshold}$, while *High_threshold* is the user-specified high threshold in dynamic load balancing, load balancing is guaranteed whenever there is a path whose maximal link utilization is below the high threshold. For the example in the figure 1, we can assure that the control for rebalancing will succeed by setting *Util_weight* =1/0.7=1.43.

In short, for dynamic load balancing, if equation (1) or (2) is used as link cost functions, which is actually proposed by [11] and [12], we can not control when rebalancing will work although much effort has been given to detect the congestion. Instead, with our proposed method, the load balancing is guaranteed once we decide to solve the congestion.

Furthermore, by assigning different *Util_weight* to individual links, policy-based routing can be easily deployed. The following is an example scenario, in one network, best-effort traffic is controlled by traffic engineering and traffic with high priorities is also routed through explicit paths over MPLS. To give preference to the traffic with high priority, we want such traffic to take the minimal hop paths. In this case, assigning large *Util_weight* to the links on minimal hop paths will keep these paths lightly loaded by best-effort traffic so that the bandwidth on minimal hop path is saved for the traffic with high priorities. Just for an example, assigning *Util_weight* =1000 to one link will prune this link from routing for best-effort traffic.

IV. CONCLUSION

In traffic engineering over MPLS, the tradeoff between limiting hop count and preferring load balancing is always the problem for path selection. Due to the different natures of the networks under control, the network carriers should be able to apply their own strategies when dealing with such problem instead of just using the mathematical solutions as in the common approaches. In this paper, we present our Strategy-based Path Selection method to provide the capability to manipulate such tradeoffs based on the network status, service requirements as well as the administrative policies. Our method is to simply add a scaling factor to link utilizations when calculating the link costs in shortest-distance routing algorithm. And by setting the different values for the scaling factor, our method is both beneficial for Quality-of-Service routing for traffic with performance requirements and dynamic load balancing for best-effort traffic.

REFERENCES

[1] RFC 2702: "Requirements for Traffic Engineering Over MPLS".

[2] S. Phuvoravan, K-T Kuo, T.Guven, L. Sudarsan, H.S. Chang, S. Bhattacharjee, and M. A. Shayman, "Fast Timescale Control for MPLS Traffic Engineering," Technical Report CS-TR-4351 and UMIACS-TR-2002-31, University of Maryland, College Park, 2002.

[3] Heeyeol Yu, Shirshanka Das, and Mario Gerla, "A Practical QoS Network Management System Considering Load Balancing of Resources," ICOIN 2003, pp. 493-503, February 2003.

[4] A. Shaikh, J. Rexford, and K. Shin, "Evaluating the Overheads of Source-Directed Quality-of-Service Routing," Proc. IEEE International Conference on Network Protocols (ICNP '98), Austin, TX, pp. 42-51, October 1998.

[5] Q. Ma and P. Steenkiste, "On Path Selection for traffic with bandwidth guarantees," In Proceedings of IEEE International Conference on Network Protocols, Atlanta, GA, October 1997.

[6] Q. Ma and P. Steenkiste, "Quality-of-Service Routing for Traffic with Performance Guarantees," In IFIP Fifth International Workshopon Quality of Service, pp. 115-126, NY, May 1997.

[7] Karol Kowalik and Martin Collier, "ALCFRA-A robust routing algorithm which can tolerate imprecisenetwork state information," Proceedings of 15th ITC Specialist Seminar.

[8] Anil Kamath, Omri Palmon, and Serge Plotkin, "Routing and admission control in general topology networks with Poisson arrivals," In Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, Atlanta, Georgia, pp. 269-278, January 1996.

[9] Stefan Butenweg, "Two distributed reactive MPLS Traffic Engineering mechanisms for throughput optimization in Best Effort MPLS networks," IEEE Symposium on Computers and Communications, July 2003, Eingesendet.

[10] Kaushik Sinha and Stephen D. Patek, "OpIATE: Optimization Integrated Adaptive Traffic Engineering," £ Systems and Information Engineering Technical Report: SIE-020001, Revised 11/12/02, Department of Systems and Information Engineering University of Virginia.

[11] H. Yamada, K. Nakamichi, M. Fukazawa and M. Katoh, "Dynamic Traffic Engineering for Network Optimization - Architecture and Evaluation," Proceedings of APNOMS2002 (The 6th Asia-Pacific Network Operations and Management Symposium), pp. 270-281, Jeju, Korea, September 2002.

[12] A. Elwalid, C. Jin, S. Low, and I. Widjaja, "MATE: MPLS adaptive traffic engineering," In Proceedings of IEEE INFOCOM 2001. pp. 1300-1309, Anchorage, Alaska, April 2001.

[13] Shyam Subramanian and Muthukumar Venkatesan, "Alternate Path Routing Algorithm for Traffic Engineering in the Internet," ITCC 2003.

[14] Boutaba, W. Szeto and Y. Iraqi, "DORA: Efficient Routing Algorithm for MPLS Traffic Engineering," International Journal of Network and Systems Management, special issue on Traffic Engineering and Management, Vol. 10, No. 3, pp. 311-327, 2002.

[15] RFC 2386: "A Framework for QoS-based Routing in the Internet".