Traffic Routing in MPLS Networks Based on QoS Estimation and Forecast

Tricha Anjali and Caterina Scoglio

Dept. of Electrical and Computer Engineering
Illinois Institute of Technology, Chicago, IL 60616
Email: tricha@ece.iit.edu

Broadband and Wireless Networking Laboratory
School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332
Email: caterina@ece.gatech.edu

***Abstract*— Emerging applications demand strict Quality of Service from the network. To meet these demands, a suitable path with enough resources needs to be selected. This paper focuses on the path computation in an MPLS network for a traffic flow, based on distance, bandwidth and delay constraints. The goal of the paper is to find a feasible path that minimizes the cost incurred. The cost is attributed to bandwidth carriage, and switching and signaling efforts in the network for the requested connection.**

***Index Terms*— Routing, QoS, MPLS, Traffic Engineering Automated Manager**

I. Introduction

To support Quality of Service (QoS) in a MultiProtocol Label Switching (MPLS) network, many algorithms have been designed for various network components, such as admission control, routing etc. However an efficient and scalable algorithm for QoS routing of traffic flows is still missing, even if much research has been focused on this subject. Usually a request requires more than one metric to be considered for routing purposes. In this sense, multi-constrained routing deals with finding a path that satisfies multiple QoS constraints on diverse metrics. It is well known that this problem is NP-complete [1], so finding an accurate and simple heuristic is one important characteristic of an efficient QoS routing algorithm. Another important issue to be addressed in the design of a QoS routing algorithm is the presence of inaccurate global network state information. Thus, an effective QoS routing algorithm must be able to work properly even while using inaccurate network information.

QoS routing is an extensively studied subject [2]. It has come a long way from the simple Dijkstra routing. Much of the work in the field of QoS routing has concentrated on the delay constrained least cost problem [3], [4]. Since the problem is NP-complete, the proposed solutions are heuristic in nature [5]. Some effort has also concentrated towards heuristic algorithms based on Lagrangian relaxation [6], [7]. The relaxation approach is based on an aggregate weight which is used in the Dijkstra algorithm for route computation. This approach does not have the capability to consider non-additive metrics. In MPLS networks, the routing research has concentrated on Label Switched Path (LSP) routing *i*.e. how to route the LSPs in the network. Many schemes such as Minimum Interference Routing Algorithm (MIRA) [8], Profile Based Routing (PBR) [9] along with modifications to OSPF, IS-IS have been proposed for LSP routing. However, to the best of our knowledge, a scheme for routing of traffic flows in an MPLS network is not considered.

This work was supported by NSF under award number ANI-0219829

In this paper, a QoS routing algorithm for traffic flows in MPLS networks is presented. This routing algorithm is unique because of the dynamic nature of the MPLS network topology. The algorithm considers multiple metrics, is scalable and operates in the presence of inaccurate information. Numerous path choices are compared in terms of their operational costs. The cost structure is defined in Section II and the cost considers all the metrics important for the path selection. The factors pertaining to the different metrics are weighed by their corresponding importance factor which can be varied from network to network. In essence, the novelty of the proposed algorithm lies in the cost structure for the LSPs and the ability to deal with the partial network state information. The proposed algorithm will be used for traffic flow routing in an MPLS network managed by Traffic Engineering Automated Manager (TEAM) [10]. TEAM is a centralized manager for automated management of a DiffServ-based MPLS network. Separating traffic belonging to different DiffServ classes over separate LSPs leads to virtual MPLS networks for each class which can be independently managed. TEAM maintains updated information about the state of the entire network for efficient, accurate management. TEAM measures various network statistics such as available bandwidth, delay, etc.

The rest of this paper is organized as follows. The network model, problem formulation and the path selection algorithm is presented in Section II. In Section III, the details of the prediction procedure to deal with presence of partial network information are presented. The performance evaluation of the proposed algorithm along with comparisons with other path selection algorithms is given in Section IV, followed by the conclusions in Section V.

II. Problem Formulation

In this formulation, the problem of QoS routing of traffic flows in MPLS networks is being considered. The goal of QoS routing is to find a low-cost feasible path that has enough available bandwidth, while restricting the number of hops and the delay on the path. The selection of the metric in each segment of the path is based on cost minimization.

We represent the network with a directed graph $G(N, L)$. Here, N is the set of nodes in the network and L is the set of LSPs. Each $l_{ij} \in L$ corresponds to an LSP between the nodes i and j $(i, j \in N)$ and is assigned a cost C_{ij}^l. A *path* p_{uv} between the nodes u and v $(u, v \in N)$ is defined as a concatenation of LSPs $l_{ux}, \ldots, l_{zv}$, where the nodes $x, \ldots, z$ are arbitrary nodes in the network. These nodes are called the *relay nodes* on the path p_{uv}. When the path p_{uv} coincides with just one LSP, we call p_{uv} as a direct path between nodes u and v and the functionalities of the relay node are not needed. There will be many paths p_{uv} between any node pair u and v, including the direct path. Let $\mathcal{P}_{uv}$ denote the set of all such paths. We associate a cost C_{uv}^p with a path p_{uv}. We use the notation $l_{ij} \in p_{uv}$ to denote that LSP l_{ij} belongs to the path p_{uv}. We define:

- A_{ij}^l : Available capacity on LSP l_{ij}
- d_{ij}^l : Delay incurred on LSP l_{ij}
- A_{uv}^p : Available capacity on path p_{uv}
- d_{uv}^p : Delay incurred on path p_{uv}
- n_{uv}^p : Number of LSPs in path p_{uv}

A_{uv}^p is the minimum of the available capacities A_{ij}^l of all the LSPs comprising the path p_{uv}. We let the path delay to be equal to the sum of the delays on the individual LSPs in the path. In other words,

$$A_{uv}^p = \min_{l_{ij} \in p_{uv}} A_{ij}^l$$

$$d_{uv}^p = \sum_{l_{ij} \in p_{uv}} d_{ij}^l$$

We need a framework to compare and choose among all the feasible paths between a node pair. Towards this end, we define the costs associated with the LSPs and paths. The cost is attributed to five factors: bandwidth requested, switching, signaling, remaining available bandwidth and delay.

The rate at which the bandwidth cost is incurred on an LSP l_{ij} depends linearly on the bandwidth required by the connection. Thus, W_{ij}^b, the bandwidth component of the cost can be written as

$$W_{ij}^{\text{b}} = b\, c_{\text{b}}^l\, T \tag{1}$$

where c_{b}^l is the bandwidth cost coefficient per capacity unit (c.u.) in the network, b is the bandwidth requested by the connection and T is the time duration for which the connection is valid. The rate of the switching cost on the LSP is also proportional to the requested bandwidth. Thus, the switching cost component can be written as

$$W_{ij}^{\text{sw}} = b\, c_{\text{sw}}^l\, h\, T \tag{2}$$

where c_{sw}^l is the switching cost coefficient per capacity unit (c.u.) in the network and h is the length of the LSP. This component is also proportional to the duration of the connection because the switching cost has to be paid for each packet of the connection as long as it holds. On the other hand, the signaling cost is a one-time cost to signal the setup of the path over the LSP. Thus, the signaling cost is given as

$$W_{ij}^{\text{sign}} = c_{\text{sign}}^l \tag{3}$$

where c_{sign}^l is the signaling cost coefficient in the network. This value is independent of the amount of bandwidth requested as it corresponds to the signaling effort, which is performed for connection establishment. Also, it is independent of the connection duration because this cost is incurred during the connection establishment. The next factor contributing to the cost is the available bandwidth left on the LSP after the connection has been granted. This cost is given as

$$W_{ij}^{\text{AB}} = \frac{c_{\text{AB}}}{A_{ij}^l - b}\, T. \tag{4}$$

Such an inverse structure is chosen for the available bandwidth cost since the available bandwidth is not a linearly additive metric (like hop, delay) in the network and LSPs with less available bandwidth are assigned a higher cost. The last term in the LSP cost comes from the delay incurred on the LSP. This cost is given as

$$W_{ij}^{\text{d}} = c_{\text{d}}\, d_{ij}^l\, T. \tag{5}$$

Summing up all these individual costs, we can obtain the total cost incurred for successfully granting the requested bandwidth on the LSP l_{ij}. However, we propose the use of weighting factors for these costs to modify the importance given to the components. A higher weighting factor would imply a higher relative significance of the associated cost component. Thus, the total cost is given as:

$$W_{ij}^l = \alpha \{ W_{ij}^{\text{b}} + W_{ij}^{\text{sw}} + W_{ij}^{\text{sign}} \} + \beta\, W_{ij}^{\text{AB}} + \gamma\, W_{ij}^{\text{d}} \tag{6}$$

Notice that we have put a single weighting factor for the first three cost components. This is because all three of them relate to the distance metric and should be weighed identically.

A path in the network is a concatenation of LSPs. If the path includes just one LSP, its cost is equal to the LSP cost. However, if the path is composed of two or more LSPs, its cost is not just the sum of the individual LSP costs. This is because the relay nodes between the LSPs have to perform additional switching and signaling due to the change in the encapsulation from one LSP to the other. Thus, the path cost is given as:

$$W_{uv}^p = \sum_{l_{ij} \in p_{uv}} W_{ij}^l + R(c_{\text{sw}}^p\, T\, b + c_{\text{sign}}^p). \tag{7}$$

Here, c_{sw}^p and c_{sign}^p denote respectively the coefficients for the IP switching and signaling costs incurred due to the presence of each relay node in the path and there are R relay nodes.

We have introduced the cost coefficients in the cost definitions above to provide a relative weight to each of the cost components. A network operator can decide these coefficients based on the fraction of the total cost that is attributed to each cost component. A study to assign values to these cost coefficients based on network characteristics is out of the scope of this paper. However, in the section on performance evaluation (Sec. IV), we have assigned values to these coefficients that we deemed appropriate.

With the above definitions of the costs involved in the path selection algorithm, we now proceed to the specification of the path selection algorithm. As mentioned before, the objective of QoS routing is to find a feasible path with enough available bandwidth while satisfying the delay and hop constraints. In our algorithm we try to achieve a balance between maximizing available bandwidth and minimizing the number of hops and delay. With this in mind, we proposed the above cost definitions for the costs.

The exact path selection problem can be specified as:

$$p^\star_{uv} : W^{p^\star}_{uv} = \min_{p \in \mathcal{P}_{uv}} W^p_{uv} \quad (8)$$

subject to the feasibility constraints $n^{p^\star}_{uv} \leq k$, $d^{p^\star}_{uv} \leq d_{\max}$ and $A^{p^\star}_{uv} \geq A_{\min}$. We have allowed a concession of k units in the length of the chosen path w.r.t. the direct LSP to be able to consider paths which are a few hops longer than the shortest path. We denote by $d_{\max}$ the maximum allowed delay and by $A_{\min}$ the minimum required available bandwidth on the path by the flow. We assume that $A_{\min}$ is larger than the bandwidth requested by the flow. This is based on two reasons. Firstly, we do not want the LSPs to get fully occupied. Secondly, since we are assuming partial information about the LSP state, the current actual values of A^l_{ij} may not be the most recently advertised value. Thus, the cushion in $A_{\min}$ over the bandwidth requested by the flow is used to compensate for the information uncertainty.

The proposed algorithm provides a heuristic to the exact path selection procedure by limiting the number of paths to be considered to F instead of an exhaustive search. We use a combination of various metrics for the path selection. The cost structure defined earlier provides a framework to choose the most efficient path for the traffic flow.

The operation of the algorithm is as follows. The centralized manager TEAM tries to find F paths between the source and the destination. These F paths are obtained by increasing the number of relay nodes in the path. In other words, if there is a direct LSP between the source and destination, it is a candidate for consideration. Next, all paths with 2 LSPs between the source and destination are considered. If such paths exceed F-1 in number, then the first F-1 paths are randomly chosen to be candidates. Note that these paths have been found without any consideration for feasibility. If still F candidates are not found, then the search proceeds to include paths which have 3 LSPs. The search goes on in this manner by increasing the number of LSPs in the path, till F candidate paths are found. These F paths are then checked for feasibility against the constraints specified in Eq. 8. The feasible paths are then the final set of candidates for routing the traffic flow. The total cost (defined in Eq. 7) of these paths is then evaluated and compared. The least cost feasible path is then chosen for the traffic flow.

This algorithm assumes knowledge of the exact values of the metrics associated with all the LSPs in the network. However, in reality, the metric updates are not instantaneous for large networks. The finite update time can compromise the scalability of the proposed routing algorithm. However, the algorithm can be made scalable by modifications that can operate in the presence of inaccurate/partial information about the network state and still providing comparable performance. In the following section, we present our approach to deal with this situation of partial information, finally leading to scalable routing algorithms.

III. Partial Information

Each network node floods information about its state to the whole network at periodic time intervals. This results in partial information about the network state at each node. The information is partial because the network nodes do not have current information about the complete network state. Instead, they have information for the time instant when the last update was generated. When a path selection request arrives, we propose to use an estimation and forecast algorithm to obtain more accurate information about the current network state. This algorithm can be applied to estimate and forecast the available bandwidth as well as delay of an LSP.

We denote by p the number of past samples that are used in the prediction and L the metric that we are trying to predict. We assume that the arrival of the path selection request is not synchronous with the update period. This means that the time interval between the instant at which the estimation is required and the arrival of last update is less than the update periodicity. So, at the instant of estimation and forecast, we have the past p samples $\{L_1, L_2 \ldots, L_p\}$ and we want to forecast the next sample L_{p+1}. We formulate the problem as a linear prediction:

$$L_{p+1} = \sum_{n=1}^{p} L_n \, w_n \quad (9)$$

where on the right side are the past samples and the prediction coefficients w_n and on the left side, the predicted value. We can rewrite the formulation as $L_{p+1} = \mathbf{L}\,\mathbf{w}^T$, where $\mathbf{L} = [L_1, L_2, \ldots, L_p]$ and $\mathbf{w} = [w_1, w_2, \ldots, w_p]$. The problem can be solved in an optimal manner using covariance method. We propose to dynamically change the value of p and based on the forecast performance. This is what distinguishes our forecast method from other linear regression schemes.

The covariance equations are given in a matrix form as $\mathbf{R}_L\mathbf{w} = \mathbf{r}$ where

$$\mathbf{R}_L = \begin{bmatrix} r_L(0,0) & \cdots & r_L(0,p-1) \\ \vdots & \ddots & \vdots \\ r_L(p-1,0) & \cdots & r_L(p-1,p-1) \end{bmatrix}$$

and $\mathbf{r} = [r_L(0,-1)\; r_L(1,-1)\; \cdots\; r_L(p-1,-1)]$ and $\mathbf{w} = [w_1\; w_2\; \cdots\; w_p]$. In order to derive the covariance from the available measurements, we estimate it as $r_L(n,m) = \sum_{i=p-N}^{p} L_{i-n}L_{i-m}$ where N affects the accuracy of the estimation. The solution of the covariance equation will provide the vector $\mathbf{w}$ that can be used for predicting $\hat{L}_{p+1}$ using Eq. 9. Now we check the location of the time instant k at which the path information is desired. If the instant k is closer to the instant p than to p+1, then we report the value L_p as the metric estimation at instant k. However, if k is closer to $p+1$, then we report the estimated value as $\hat{L}_{p+1}$. To further improve the prediction performance, we propose to use an update algorithm to adjust the value of p, the number of past samples considered in the prediction. According to this algorithm, if the ratio of the prediction error e to the actual value L_{p+1} is above a threshold e_{Th}, which implies that the error is large w.r.t. the actual value and the prediction performance was not too good,

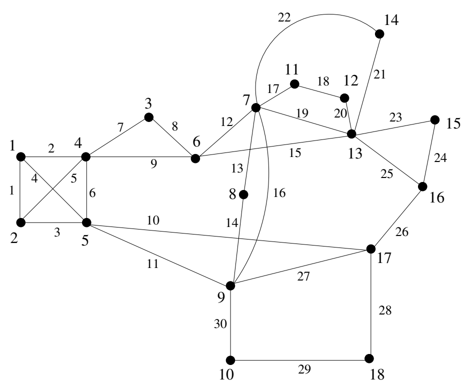

Fig. 1. Network Topology.

we increase the value of p to consider more samples in the prediction process. If the ratio is smaller than e_{Th}, then we reduce the value of p. However, we put an upper bound p_{max} on p because large values of p increase the computational cost of the regression. The threshold is determined based on the traffic characteristics and the conservatism requirements of the network domain. It represents the confidence in the estimation procedure in terms of prediction errors. By using this prediction approach, we are eliminating the drawback of the periodic update approach related to the unresponsiveness to significant metric changes.

IV. Performance Evaluation

In this section we compare and evaluate the proposed path selection algorithm, along with other well-known algorithms, from the viewpoint of performance and robustness. We conducted extensive simulations to evaluate the performance and computational complexity of the proposed algorithmic solution. The goal of these simulations is to evaluate the goodness of the proposed algorithm and to demonstrate the benefits of the approach of using multiple metrics for path determination without loss of routing performance.

We used the topology of Fig. 1 for our simulation experiments. This represents a popular "isp" topology used in many QoS routing studies. All the links are bidirectional with a capacity of 155 c.u. in both directions. We set k, the number of extra hops allowed in the feasibility constraint of the path selection problem in Eq. 8, to 5 in order to allow for paths longer than the shortest path. We restrain the value of A_{min} in the path selection problem to be at least 10 c.u. above the bandwidth requested. We restrict the number of paths considered for feasibility check and cost comparison to F, which is set to 20. This value was chosen so that we have enough number of paths that are distinct from the min-hop path, and at the same time we do not perform an exhaustive search for the paths. We assigned unity values to each of the cost coefficients and also to the three weighting factors α, β and γ in the cost formulation of Section II. Cost coefficients are special quantities that vary from network to network depending on the parameters important to the network. By suitable choice of the weighting factors, we can obtain the routing performance of several well-known routing algorithms. The weighting factors can be adapted to the traffic load in the network. For example, if the network is lightly loaded, the shortest path routing can give a satisfactory performance. The shortest path algorithm can be obtained by setting $\alpha = 1$ and $\beta = \gamma = 0$. Since we do not know the network parameters and the network conditions, we assign unity values to the cost coefficients. In this way, we are impartial to each cost component. The weighting factors in the cost formulation were also assigned unity values to give equal importance to each cost metric.

We performed 25 independent experiments with same traffic profile. We introduced traffic from nodes on the left side of the network (1,2,4,5) towards the right side (11, 12, 13, 14, 15, 16, 17). In this way, we have introduced focused overload in the middle of the network. In such a scenario, using shortest path routing algorithm can be penalizing as the network is overloaded. Thus, a more intelligent and efficient routing algorithm should be preferred which will give a better performance. This is confirmed by the results of Fig. 2. We present the rejection ratio of the proposed algorithm in comparison to the shortest path routing algorithm. As can be seen, our algorithm has reduced the rejection ratio by around 75% w.r.t. the shortest path routing.

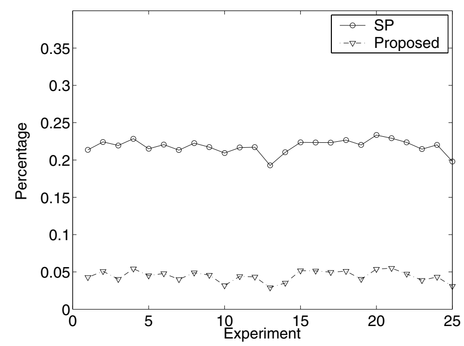

Fig. 2. Rejection ratio.

Next, we compare the performance w.r.t. the metrics we have considered. We present the minimum available bandwidth among all the LSPs in the network. These results are presented in Fig. 3.

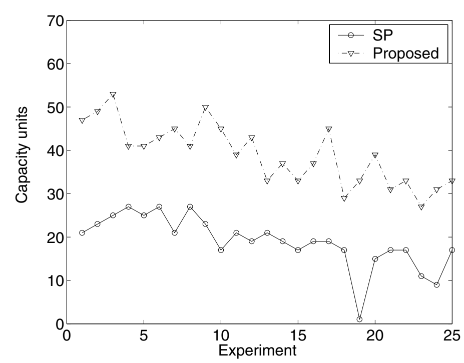

Fig. 3. Minimum available bandwidth.

As can be seen, the minimum available bandwidth is lower for the shortest path routing in contrast to the proposed algorithm. This is expected because the shortest path routing is limited to only one path between a node pair for every request, unlike the proposed algorithm that selects paths by considering a combination of metrics. On the other hand, the mean available bandwidth is larger for the shortest path routing. This gives the false impression that the performance of the shortest path algorithm is better than our algorithm. However, this is attributed to the poor load balancing achieved by the shortest path routing, as opposed to our algorithm. The shortest path algorithm chooses the same path between a node pair every time it is executed. Thus, it has a high rejection ratio and the load is concentrated on a few LSPs. Our algorithm chooses possibly different paths for the requests (depending on the network state) and thus distributes the load in the network achieving a lower rejection rate.

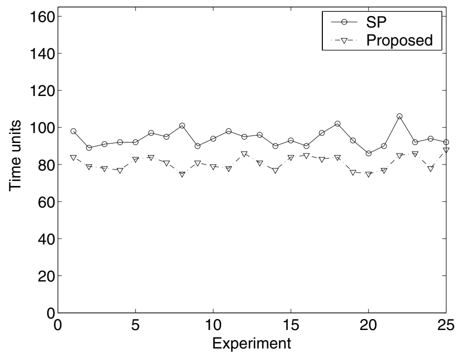

Fig. 4. Minimum delay.

The second metric we compare is the delay encountered by the request packets. The delay is composed of three components: the transmission, propagation and the queuing. We assume that the first two components are constant, however the queuing delay is determined by the load on the LSP. In other words, the queuing delay is larger for a highly loaded LSP than a lightly loaded LSP. In Fig. 4, we present the results for the delay encountered in the network. The minimum delay incurred by the shortest path routing is larger than the proposed algorithm. This is due to the over-loading of a few LSPs in the network.

The third metric is the number of paths with relay nodes. This number is obtained by taking a network snapshot at some time. We count the number of requests that were routed along paths with relay nodes. We show the plot in Fig. 5. Obviously, the number of paths with relay is 0 for the shortest path routing whereas the proposed routing algorithm has a large number of paths with relay nodes.

With these results, we have compared the performance of the proposed algorithm with the shortest path routing algorithm. We chose the shortest path algorithm as the basis for performance comparison because it is the current routing scheme in the Internet. By choosing appropriate values for the weighting factors α, β, γ in our algorithm, we can obtain other routing schemes and their results. We found that the performance of the proposed algorithms is superior to the shortest path routing. However, this is achieved at the expense of increased computational complexity. The proposed algorithm needs state information for the whole network and it should be updated periodically.

V. Conclusions

In this paper, we have presented a routing algorithm to find feasible paths that minimize the cost incurred by an MPLS network to support the user bandwidth requests. The cost is attributed to bandwidth carriage, and switching and signaling efforts in the network for the requested connection. The routing algorithm is scalable and operates under inaccurate network information. A prediction algorithm is utilized to cope with partial network information. The performance of the algorithm was compared to the shortest path routing algorithm. We plan to extend the paper by finding a method to dynamically assign the cost coefficients and weighting factors in the cost formulation to adapt the routing to the network state.

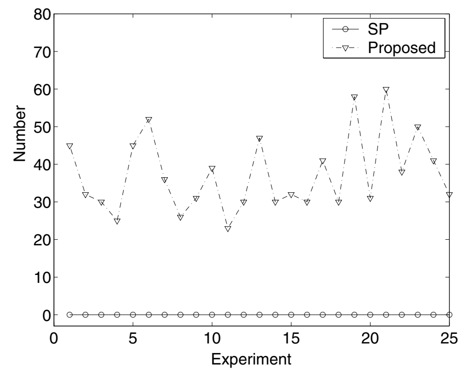

Fig. 5. Number of paths with relay nodes.

References

[1] Z. Wang and J. Crowcroft, "Quality-of-Service Routing for Supporting Multimedia Applications," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 7, pp. 1288–1234, September 1996.

[2] S. Chen and K. Nahrstedt, "An Overview of Quality of Service Routing for the Next Generation High Speed Networks: Problems and Solutions," *IEEE Network*, vol. 12, no. 6, pp. 64–79, 1998.

[3] G. Feng, C. Douligeris, K. Makki, and N. Pissinou, "Performance Evaluation of Delay-Constrained Least-Cost QoS Routing Algorithms Based on Linear and Nonlinear Lagrange Relaxation," in *Proceedings of IEEE ICC'02*, New York, USA, April 2002, pp. 2273–2278.

[4] D. S. Reeves and H. F. Salama, "A Distributed Algorithm for Delay-constrained Unicast Routing," *IEEE/ACM Transactions on Networking*, vol. 8, no. 2, pp. 239–250, April 2000.

[5] X. Yuan, "Heuristic Algorithms for Multi-Constrained Quality of Service Routing," *IEEE/ACM Transactions on Networking*, vol. 10, no. 2, pp. 244–256, April 2002.

[6] A. Jüttner, B. Szviatovszki, I. Mécs, and Z. Rajkó, "Lagrange Relaxation Based Method for the QoS Routing Problem," in *Proceedings of IEEE INFOCOM'01*, Anchorage, USA, April 2001, pp. 859–868.

[7] T. Korkmaz and M. Krunz, "Multi-constrained Optimal Path Selection," in *Proceedings of IEEE INFOCOM'01*, Anchorage, USA, April 2001, pp. 834–843.

[8] K. Kar, M. Kodialam, and T. V. Lakshman, "Minimum Interference Routing of Bandwidth Guaranteed Tunnels with MPLS Traffic Engineering Application," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 12, pp. 2566–2579, December 2000.

[9] S. Suri, M. Waldvogel, and P. Warkhede, "Profile-Based Routing: A New Framework for MPLS Traffic Engineering," in *Proceedings of 2 International Workshop on Quality of future Internet Services (QofIS'01)*, Coimbra, Portugal, September 2001, pp. 138–157.

[10] I. F. Akyildiz, T. Anjali, L. C. Chen, J. C. de Oliveira, C. Scoglio, A. Sciuto, J. A. Smith, and G. Uhl, "A New Traffic Engineering Manager for DiffServ/MPLS Networks: Design and Implementation on an IP QoS Testbed," *Computer Communications*, vol. 26, no. 4, pp. 388–403, March 2003.