A Scalable Algorithm for Link-state QoS-based Routing with Three Metrics

Luís Henrique M. K. Costa[1,2] Luis.Costa@lip6.fr

Serge Fdida[1] Serge.Fdida@lip6.fr

Otto Carlos M. B. Duarte[2] otto@gta.ufrj.br

[1] *LIP6 – Université Pierre et Marie Curie – 4, place Jussieu - 75252 - Paris Cedex 05 - France*

[2] *GTA/COPPE/EE – Universidade Federal do Rio de Janeiro – C.P. 68504 - 21945-970 - Rio de Janeiro, RJ - Brasil*

Abstract— **Quality of Service (QoS) based routing provides QoS guarantees to multimedia applications and an efficient utilization of the network resources. Nevertheless, QoS routing is likely to be a costly process that does not scale when the number of nodes increases. Thus, the routing algorithm must be simple. This paper proposes and analyzes the performance of a link-state QoS routing algorithm that takes into account three metrics: propagation delay, available bandwidth, and loss probability. A heuristic based on the residual loss probability and metric-combination is used to turn the algorithm scalable and solvable in polynomial time. The simulation results show that our algorithm is a promising solution to construct paths constrained on three metrics.**

I. Introduction

The next generation of the Internet will offer new classes of service to provide Quality of Service (QoS) guarantees to applications. An important issue for that purpose is the definition of a routing architecture that considers the QoS requirements of applications [1]. In traditional routing, packets are delivered using a route based on their source and destination addresses while in QoS-based routing the traffic requirements are also taken into account by the routing algorithm.

In classical routing, one single metric is used in route decisions. QoS-based routing is more complex because the decisions are based on multiple metrics. Hence, scalability becomes a main issue [1], [2]. The routing algorithm must be simple. Some authors consider the use of pre-computed paths to minimize the setup overhead [2], [3]. The uncertainty of QoS parameters may also affect QoS routing, e.g., in a link-state protocol the network database can be out of date [4], [5]. Setting the period of updates is a tradeoff between the accuracy of QoS parameters and the protocol overhead [2], [6].

Per-flow state should be avoided as this approach is unscalable [7], [8]. A better solution is to use per-flow state only for special flows (that have very stringent requirements). Less demanding flows can be satisfied with a connectionless service, where routing is done per class of service instead of per flow. Combining the class-of-service approach with per-flow routing for few flows may increase the routing efficiency. The work in [9] treats the problem of multicast routing with resource reservations and analyzes this combination.

This work was sponsored by FUJB, CNPq, CAPES/COFECUB, and IST Project GCAP N^0 1999-10504.

This paper proposes SMM-LS (Single Mixed Metric - Link-State routing) an algorithm for unicast route computation based on link-states. SMM-LS computes routes with three metrics: propagation delay, available bandwidth and link loss probability. The SMM basic idea was first presented in [10]. The present paper proposes a different algorithm that improves the performance of the original proposal. The performance of SMM-LS was evaluated through simulation. The performance analysis compares SMM-LS with some other approaches that are also based on Dijkstra's shortest-path algorithm. The objective is to evaluate the call acceptance probability of the proposed algorithm. The results show that SMM-LS is a promising algorithm for three-metric route computation.

This paper is organized as follows. Section II presents the SMM-LS proposal, Section III presents the simultation results and Section IV concludes the paper.

II. The SMM-LS Algorithm

In [11] we proposed and analyzed the performance of SMM - Distance Vector (SMM-DV) algorithm for hop-by-hop routing. In this paper we propose an improved version of SMM based on link-states (SMM-LS). This algorithm assumes that routes are fixed after connection setup using some kind of explicit routing, pinning, or source routing.

QoS routing algorithms may treat the metrics in different ways. First, the metrics may be dependent or not. End-to-end transmission delay depends on the link capacity, reserved bandwidth, propagation delay, and loss probability. Ma and Steenkiste [12] proposed algorithms based on dependent metrics but defining their relationship is a major problem because it depends on the traffic model.

The metrics can also be independent from each other. For instance, propagation delay is independent from the available bandwidth. In this case, the computational complexity of the algorithm is the main problem as discussed in [10]. The proposals found in the literature treat independent metrics in one of three ways: as an overall cost, individually, or by means of heuristics. An overall cost is a function such as

$$f(p) = \frac{B(p)}{D(p) \times L(p)} \tag{1}$$

where $B(p)$ is the available bandwidth, $D(p)$ the propagation delay, and to $L(p)$ the loss probability. The overall cost has two problems. First, maximizing $f(p)$ does not guarantee each of the QoS requirements. Second, the composition rule of this function is not clear since the composition rules of its parameters are different.

Treating the metrics individually is not feasible due to the algorithm complexity. The problem of finding a path with n additive and m multiplicative metrics is NP-complete if $n + m \geq 2$ [13]. There are different heuristic proposals for QoS routing [14], [15], [16], [17]. The most performing heuristics in terms of network resource utilization suppose centralized route computation, not suitable for large networks.

SMM-LS uses independent metrics. The key idea is to execute a search in the graph to eliminate all arcs that do not have enough bandwidth and then to run a variant of Dijkstra's algorithm. SMM-LS uses a single mixed metric of delay and loss probability. Instead of loss probability, the absolute value of the logarithmic transmission-success probability function ($slog(p) = |log(1 - L(p))|$) is used to avoid complex composition rules.

Furthermore, we assume that routes can not have more than 90% of loss probability, so $0 < |slog| < 1$. As ms unit is appropriate [18] to represent delay, we assume that delay is an integer. As a consequence, we have a simple single-metric representation of delay and loss, where the integer part is delay and the decimal part represents loss. This property allows the construction of a simple heuristic.

Let $G = (N, A)$ be a network with N nodes and A arcs. Each arc (i, j) is assigned two real numbers: b_{ij} is the available bandwidth and $u_{ij} = slog_{ij} + d_{ij}$ is the single mixed metric, where $slog_{ij}$ is the logarithmic transmission-success probability and $d_{i,j}$ is the propagation delay. When arc (i, j) is inexistent $b_{ij} = 0$ and $u_{ij} = \infty$. Given a path $p = (i, j, k, ..., q, r)$, the path width, $w(p)$, is $min[b_{ij}, b_{jk}, ..., b_{qr}]$ and its length is $l(p) = u_{ij} + u_{jk} + ... + u_{qr}$. Given three constants, $\mathbf{B}$ the minimum bandwidth, $\mathbf{D}$ the maximum delay, and $\mathbf{P}$ the maximum $slog$ ($\mathbf{U} = \mathbf{D} + \mathbf{P}$ is the maximum single mixed metric), the problem is to find a path, p, between i and r such that $w(p) \geq \mathbf{B}$ and $l(p) \leq \mathbf{U}$.

Given a path p with x hops, if $slog_{ij} < \mathbf{P}/x$ for each link (i, j) in p then $slog(p) < \mathbf{P}$. Nevertheless, x is unknown during the path computation. The idea is then to restrict $slog_{ij}$ to $\mathbf{P}/C$, where C is the length of the longest path on the graph (this is a pessimistic hypothesis). If $slog_{ij} < \mathbf{P}/C$ at each step, then $slog(p) < \mathbf{P}$. This limitation may be, however, too strong because the route may have a hop-length smaller than C.

This restriction can be relaxed. At each step, a hop remainder (SS_i) and a logarithmic probability remainder (SP_i) are associated to node i. SS_i is the difference between the length of the longest path in the graph and the length of the candidate path from the source to node i. SP_i is the difference between the required limit and the present value of the logarithmic transmission success of the candidate path from the source to node i. The maximum logarithmic probability that can be added at each step is limited to SP_i/SS_i instead of $\mathbf{P}/C$. This value can be larger than $\mathbf{P}/C$, in the case where SP_i is smaller than $SP_i \times (\mathbf{P}/C)$. So the probability of finding a path in the network graph grows. Supposing that node 1 is the source, SMM-LS is as follows:

`Step 1:` $\forall i, j : u_{ij} = \infty$ `if` $b_{ij} < \mathbf{B}$.
`Step 2:` $L = \{(1, 1)\}$; $SS_1 = C$; $SP_1 = \mathbf{P}$.
  $\forall i, i \neq 1$:
    $U_i = u_{1i}$ `and` $ph_i = 1$
    `if` $(U_i == \infty)$, $SS_i = C$, $SP_i = \mathbf{P}$;
      `else` $SS_i = C - 1$, $SP_i = \mathbf{P} - (\mathbf{u_{1i}} - \lfloor \mathbf{u_{1i}} \rfloor)$.
`Step 3:` `Find` $k \notin L$ |
    $(U_k = \min_{i \notin L} U_i$ `and` $U_i - \lfloor U_i \rfloor \leq \mathbf{P})$.
  `If` $U_k > \mathbf{U}$,
    `there is no feasible path. Stop.`
  $L := L \cup \{(k, ph_k)\}$.
  `If` L `contains node` m,
    `a path was found. Stop.`
`Step 4:` $\forall i \notin L$:
  `If` $((U_k + u_{ki} < U_i)$ `and`
      $(u_{ki} - \lfloor u_{ki} \rfloor < SP_i/SS_i))$,
    `or if` $((U_k + u_{ki} < \mathbf{U})$
      `and` $(\frac{SP_k - \lfloor u_{ki} \rfloor}{SS_k - 1} < \frac{SP_i}{SS_i}))$,
    `do`
      { $U_i := U_k + u_{ki}$.
        $ph_i := k$.
        $SS_i := SS_k - 1$.
        $SP_i := SP_k - \lfloor u_{ki} \rfloor$. }
    `Go to step 3.`

Step 4 is differs from the first SMM [10]. This step checks if node k is a better previous hop to the i nodes that are not in the path being constructed (L). If choosing k provides a smaller length, than it is chosen as previous hop according to the original Dijkstra's algorithm. The second test in line 2 of Step 4 guarantees the remainder ratio of our heuristic. Line 3 provides a second possibility of choosing k as previous hop: even if the path length is larger through k, k may be chosen as previous hop if this new length is feasible and the remainder ratio is better.

Step 1 consists of $O(N^2)$ operations and is executed once. Step 2 requires $O(N)$ operations and is also executed once. Step 3 and 4 require $O(N)$ operations and are repeated $N - 1$ times in the worst case. So the complexity of the above algorithm is $O(N^2)$, the same as the original Dijkstra's algorithm.

The SMM-LS heuristic can be improved trading off additional complexity. The idea is to repeat the Dijkstra search for different values of C (say $\mathbf{C} = \{C_i\}$) , thus increasing the probability of finding a feasible path. This means that steps 2 to 4 are repeated $|\mathbf{C}|$ times. The complexity of the algorithm then turns $O(|\mathbf{C}| \times N^2)$. As long as $|\mathbf{C}|$ is small the complexity of SMM-LS is acceptable.

III. SMM-LS Simulation

We simulated SMM-LS with NS [19]. The implementation consists of two new routing modules and a modified link which admits three metrics. The Integrated Services module simulates the admission control, connection setup, and bandwidth reservation. The call blocking probability evaluates the efficiency of SMM-LS.

SMM-LS is compared to five algorithms. The first one is the classical Dijkstra's shortest-path algorithm (DSP) using the number of hops as metric. Theoretically DSP should have the worst acceptance probability since it does not consider any QoS requirement. Therefore the DSP results are used as reference to compare the other algorithms. BWDD (BandWidth pruning and Delay Dijkstra search) implements the proposal of Wang and Crowcroft [13]. BWMD (BandWidth pruning and Mixed metric Dijkstra search) is the same as BWDD, but the Dijkstra search uses a single mixed metric of delay and loss probability. SMM-LS is our proposed algorithm. OC (Overall Cost) implements a Dijkstra search based on an overall cost function. SOC (Scaled OC) is a modified version of OC, where the weights of the three metrics are scaled so as to have the same importance in the cost function.

A. Experiments

The topology used is similar to the one of [2] and typical of a large ISP's network. Link propagation delays and loss probabilities are uniformly distributed on the interval $[min, max]$ (Table I). Link bandwidth is homogeneous.

TABLE I

Characteristics of the simulated topology.

Link bandwidth	10Mbps
Number of nodes	18
Number of links	30

	min	max
Link delay	1 ms	10 ms
Link loss	0.001	0.005

Each call consists of a source-destination pair plus the delay, bandwidth, and loss probability requirements. The calls are supposed to arrive independently at each node, following a Poisson distribution. Call inter-arrival time is exponentially distributed with average $0.4s$. Call holding time is exponentially distributed with $3s$ average. QoS requirements varied in the different experiments. Each simulation lasts for 1500 seconds and is repeated 10 times, leading to approximately 35,000 connection requests for each configuration described hereafter.

In the first three experiments one QoS requirement is varied and the other are small and fixed. The objective is to evaluate the sensitivity of each algorithm to each single requirement. Table II summarizes the experiment characteristics. In the graphs that follow the X-axis represents the upper bound limit of the QoS requirement interval. The Y-axis is the call acceptance probability in the top graphs and the difference to DSP in the bottom graphs.

TABLE II

Characteristics of the connection requests.

	QoS requirements								
	Delay (ms)			Bandwidth ($\times 64kbps$)			Loss ($\times 10^{-3}$)		
Exp.	dl_i	dl_f		bw_i	bw_f		lp_i	lp_f	
		min	max		min	max		min	max
1	1	5	100	$fixed$			$fixed$		
2	$fixed$			1	10	160	$fixed$		
3	$fixed$			$fixed$			1	5	100
4	1	5	100	1	10	160	1	5	100

In Exp. 1, the bandwidth and loss probability required are negligible, so delay dominates the path choice. The delay is uniformly distributed on $[dl_i, dl_f]$. The upper bound of the interval (dl_f) is varied from min to max as shown in Table II. The larger the interval, the looser the QoS required. In Exp. 2, delay and loss probability are fixed. The bandwidth is uniformly distributed on $[bw_i \times 64kbps, bw_f \times 64kpbs]$. bw_f is varied, but in this case, the larger the interval, the stronger the QoS required. Exp. 3 has delay and bandwidth requirements fixed whereas the loss probability requested is uniformly distributed in the interval $[lp_i, lp_f]$ (Table II). Here again, as lp_f grows, the QoS requirements are looser so the call acceptance grows (Figure 3). In Exp. 4, the 3 series of QoS requests are combined (Table II). The order of the bandwidth requests is inverted so the low QoS levels in Figure 4 have strong requirements (level 0 groups the three tightest requirements whereas level 10 the loosest requests).

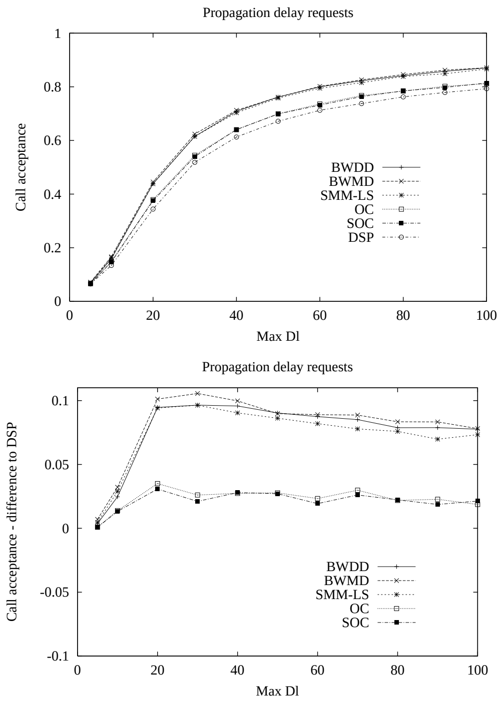

Fig. 1. Call acceptance versus delay variation.

B. Analysis

Delay requirement – Figure 1 shows the results of Exp. 1. As expected, DSP yields the worst call acceptance probability. BWDD, BWMD and SMM-LS perform near the same. This is natural, since the three approaches reduce to a delay Dijkstra search in this experiment. SMM-LS has a little disadvantage because the remainder heuristic may not consider some paths in route computation. OC and SOC present the worst performance showing that, at least for this topology and metrics, the overall cost approach is not well suited. Both OC and SOC suffer from the effect of unbalanced QoS requirements - the minimum limit required on $f(p)$ (Equation 1)) does not reflect the QoS demanded. The reason is that the same $f(p)$ may be obtained with a large $D(p)$ and small $L(p)$ or with a small $D(p)$ and large $L(p)$. The overall cost hides the relative importance of the different QoS requirements.

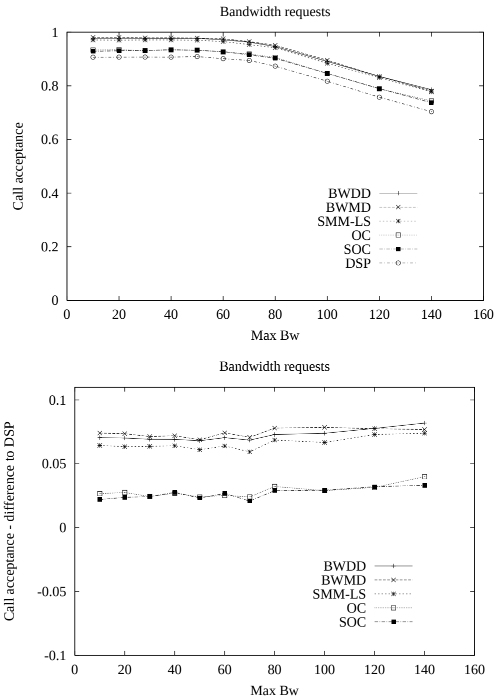

Fig. 2. Call acceptance versus bandwidth variation.

Bandwidth requirement – Figure 2 shows Exp. 2, where bandwidth is varied. DSP has the worst acceptance probability since it does not consider bandwidth in route computation. BWDD, BWMD and SMM-LS have similar performance. This is expected since the three algorithms treat the bandwidth requirement in the same way - by pruning infeasible links. Another time the single-function approaches, OC and SOC, present the worst performance. The reason for this poor performance is the same as in Exp. 1. QoS requirements are different in scale causing the overall cost to hide the real needs of the connections.

Loss probability requirement – Figure 3 presents the results from Exp. 3, where bandwidth and delay requirements are small and fixed whereas the loss probability is varied. This graph shows that the heuristic employed by SMM-LS is effective in guaranteeing the loss probability requirement. When this requirement is too tight (lp_f is small) all algorithms perform bad, even when compared to DSP. BWDD and BWMD perform nearly the same, showing that the use of the single mixed metric in BWMD is not effective. This is due to the domination of delay in the mixed metric. The performances of OC and SOC are similar, but slightly better when compared to Exps. 1 and 2. This is because the loss probability requirement is numerically small, which causes the disequilibrium among the metrics to be smaller and the overall-cost algorithms to perform better.

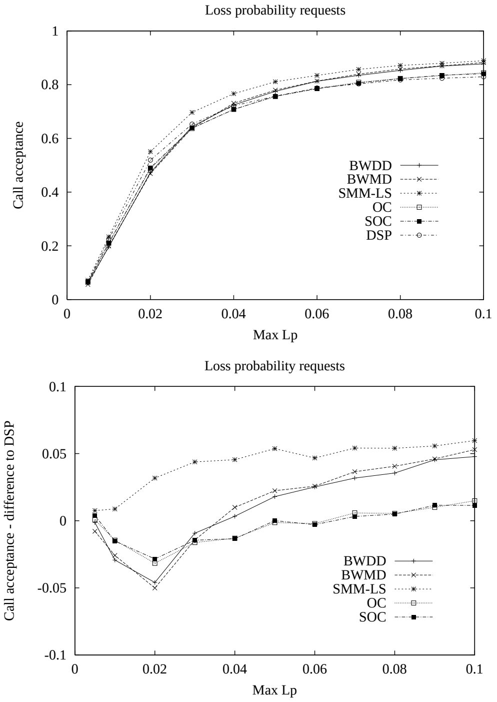

Fig. 3. Call acceptance versus loss probability variation.

Three-requirement requests – Figure 4 presents the results of Exp. 4. The first graph shows that the QoS levels in the left side are too tight, thus the call rejection is very high. The looser QoS levels yield a call acceptance probability near 80 %. When the QoS levels are very loose (e.g. QoS levels 9 and 10 in the second graph of Figure 4), BWDD and BWMD perform similar to SMM-LS. This is due to the weak loss probability requirement. On the other hand, when the QoS requirements are too tight (QoS levels 0 and 1) all algorithms perform near the same (almost all connection requests are infeasible).

The best performance of SMM-LS is in the middle of the QoS-level interval. In this band SMM-LS is able to carry a larger number of flows than all other approaches, guaranteeing a high level of network resource utilization.

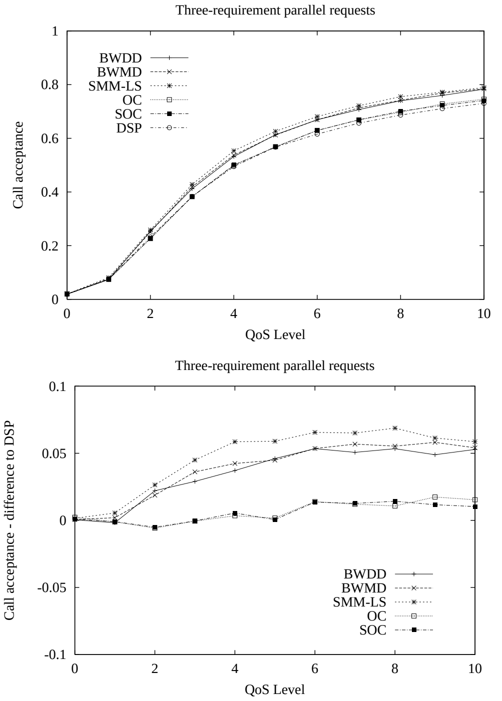

Fig. 4. Call acceptance versus QoS level variation.

It is woth noting that this higher acceptance probability is achieved through an additional processing overhead, as we explained in Section II. This processing penalty is fortunately not heavy. For the ISP topology the size of $\mathbf{C}$ equals 9. Nevertheless, $|\mathbf{C}|$ may be easily reduced, since the largest C_i's were rarely used. Reducing $\mathbf{C}$ will thus weakly impact the call acceptance probability.

IV. Conclusions

QoS routing provides better QoS guarantees to applications and improves the network resource utilization. Nevertheless QoS routing is a costly mechanism. This paper proposed SMM-LS, a unicast link-state routing algorithm capable of finding routes constrained on three QoS requirements, namely, the propagation delay, the available bandwidth, and the loss probability. SMM-LS is based on a modified version of Dijkstra's shortest-path algorithm and on the utilization of a remainder-based heuristic. We showed that the complexity of SMM-LS is compensated by the gain on call acceptance probability, which indeed leads to an efficient network resource utilization. The improvements added to the first version of SMM [10] were fundamental in achieving a higher acceptance rate.

Our future work includes the reduction of link-state information exchanged. Link-states would be modified just when resource-intensive flows arrive or leave the network. The objective is to identify the tradeoff between the accuracy of state information and the call acceptance probability. We will also analyze the suitability of SMM-LS for constructing QoS-constrained multicast trees. We will compare the performance of SMM-LS with a heuristic similar to that used in [15], [16].

References

[1] E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick, *A Framework for QoS-Based Routing.* RFC 2386, Aug. 1998.

[2] G. Apostolopoulos, R. Guerin, S. Kamat, and S. K. Tripathi, "Quality of service based routing: A performance perspective," in *ACM SIGCOMM'98*, pp. 17–28, Sept. 1998.

[3] N. Traft-Plotkin, B. Bellur, and R. Ogier, "Quality-of-Service routing using maximally disjoint paths," in *IEEE/IFIP IWQoS'99*, June 1999.

[4] D. H. Lorenz and A. Orda, "QoS routing in networks with uncertain parameters," in *IEEE INFOCOM'98*, 1998.

[5] V. Fayet, D. A. Khotimsky, and T. Przygienda, "Hop-by-hop routing with node-dependent topology information," in *IEEE INFOCOM'99*, Mar. 1999.

[6] A. Shaikh, J. Rexford, and K. G. Shin, "Evaluating the overheads of source-directed quality-of-service routing," in *International Conference on Network Protocols*, 1998.

[7] R. Braden, D. Clark, and S. Shenker, *Integrated Services in the Internet Architecture: An Overview.* RFC 1633, June 1994.

[8] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, *An Architecture for Differentiated Services.* RFC 2475, Dec. 1998.

[9] B. Rajagopalan and R. Nair, "Multicast routing with resource reservation," *Journal of High Speed Networks*, vol. 7, no. 2, July 1998.

[10] L. H. M. K. Costa and O. C. M. B. Duarte, "A scalable QoS-based routing mechanism for supporting multimedia applications," in *IEEE International Conference on Multimedia Computing and Systems - ICMCS'99*, vol. 2, pp. 346–351, June 1999.

[11] L. H. M. K. Costa, S. Fdida, and O. C. M. B. Duarte, "Distance-vector qos-based routing with three metrics," in *IFIP Networking'2000 / HPN - High Performance Networking*, May 2000.

[12] Q. Ma and P. Steenkiste, "Quality of service routing for traffic with performance guarantees," in *IFIP International Workshop on Quality of Service*, pp. 115–126, May 1997.

[13] 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. 1228–1234, Sept. 1996.

[14] R. Vogel, R. G. Herrtwich, W. Kalfa, H. Wittig, and L. C. Wolf, "QoS-based routing of multimedia streams in computer networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 7, pp. 1235–1244, Sept. 1996.

[15] W. C. Lee, M. G. Hluchyj, and P. A. Humblet, "Routing subject to quality of service constraints in integrated communication networks," *IEEE Network Magazine*, vol. 9, no. 4, pp. 46–55, July-Aug. 1995.

[16] L. L. H. Andrew and A. A. N. A. Kusuma, "Generalised analysis of a QoS-aware routing algorithm," in *IEEE GLOBECOM'98*, pp. 118–123, 1998.

[17] H. F. Salama, D. S. Reeves, and Y. Viniotis, "A distributed algorithm for delay-constrained unicast routing," in *IEEE INFOCOM'97*, 1997.

[18] A. Fei, G. Pei, R. Liu, and L. Zhang, "Measurements on delay and hop-count of the Internet," in *IEEE GLOBECOM'98 - Internet Mini-Conference*, 1998.

[19] K. Fall and K. Varadhan, *The ns Manual.* UC Berkeley, LBL, USC/ISI, and Xerox PARC, Jan. 2001. Available at http://www.isi.edu/nsnam/ns/ns-documentation.html.