Internet QoS Routing using the Bellman-Ford Algorithm

D. Cavendish and M. Gerla
Computer Science Department,
University of California at Los Angeles
405 Hilgard Avenue, Los Angeles, CA 90024-1596, USA
{dirceu,gerla}@cs.ucla.edu

Abstract

Multimedia applications are Quality of Service (QoS) sensitive, which makes QoS support indispensable in high speed Integrated Services Packet Networks (ISPN). An important aspect is QoS routing, namely, the provision of QoS routes at session set up time based on user request and information about available network resources. This paper develops optimal QoS routing algorithms within an Autonomous System (AS). Previous approaches have been based either on minimizing a **single** metric (delay, for instance) or a combination of multiple metrics, optimizing one at a time, in a hierarchical fashion. Our approach finds minimum hop paths which satisfy **multiple** QoS constraints. We argue that a QoS version of the Bellman-Ford routing algorithm provides the best strategy for QoS routing problems of a given type. We show that Bellman-Ford is very powerful in solving most multiple constrained routing problems arising in a flat network (within an autonomous system), if the minimum hop is the main objective function. We further illustrate the importance of Bellman-Ford QoS routing algorithms with regard to network utilization and session blocking through simulation experiments.

Keywords

Internet Routing, Quality of Service Constraints

1 INTRODUCTION

Quality of Service (QoS) sensitive applications are becoming popular, as Internet users move to more demanding applications with regard to the types of service requested from a packet network. Thus, QoS support for these applications is necessary to meet oftentimes stringent end-to-end requirements such as bandwidth, delay, and packet loss. The support of QoS applications in a packet network includes a broad range of functions such as priority mechanisms for flows, scheduling disciplines, traffic shaping schemes, and routing algorithms. This paper is concerned with the last issue, i.e., the provision of

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35388-3_42

H. R. van As (ed.), *High Performance Networking*

routing algorithms capable of finding routes which comply with QoS applications requirements.

Routing QoS sensitive applications consists in finding a path which complies with end-to-end constraints derived from the QoS users needs. This task translates into finding routes in a network scenario of multiple metrics. In general, routing optimally a flow with such end-to-end constraints (multiple metrics) is known to be an NP-complete problem (Garey *et al.* 1979). Recent papers have addressed the QoS constrained routing problem using a variety of strategies. Some researchers have proposed approximate solutions based on minimizing a single objective function at a time (i.e., defining a hierarchy of metrics) (Guerin *et al.* 1997a) for a generic multiple metric routing problem. Others have succeeded in reducing the problem complexity to a polynomial degree by assuming a particular scheduling discipline and traffic shaping policy at each router ((Pornavalai *et al.* 1997) (Zhao *et al.* 1997)). Our work somehow lies in between these two approaches, as it makes certain assumptions about the multiple metrics which define the problem, but not about specific router behavior in handling the traffic flow.

In considering routing with multiple constraints, it has been noticed that the Bellman-Ford (BF) algorithm can potentially solve a two metric routing problem, when one of the metrics is the hop count (Guerin *et al.* 1997a). This fact favors the Bellman-Ford algorithm when compared with the Dijkstra algorithm, which lacks this capability. This is because Dijkstra algorithm does not search for a minimum path cost in ascending order of number of hops, as Bellman-Ford does. We have decided to investigate further the Bellman-Ford algorithm's ability to solve multiple constrained routing problems. The use of a minimum hop routing strategy is justified since it is likely to provide the lowest session rejection probability for a given network load, if no knowledge about the distribution of connections is assumed. Note that QoS applications usually require end-to-end **bounds** with regard to various metrics, rather than the minimization of a specific end-to-end metric (e.g., delay). We have chosen to find the path with minimum number of hops among all which satisfy the multiple end-to-end constraints.

The paper is organized as follows. Section 2 describes the model for a flat (AS domain) network model with multiple metrics. It also describes related work in the QoS routing area. In section 3, we define a Bellman-Ford algorithm capable of handling multiple cost metrics. We show some routing problems that can be solved exactly and others which cannot be solved by this enhanced Bellman-Ford algorithm. Section 4 explains how bandwidth, buffer, delay, delay jitter, and packet loss metrics can be handled by a Bellman-Ford algorithm, in a typical QoS supportive packet network. In section 5, we exemplify the use of a multiple metric Bellman-Ford algorithm in supporting QoS flows, comparing it with other popular routing strategies. Section 5 concludes the paper and points to future research directions.

2 QOS ROUTING WITH MULTIPLE CONSTRAINTS

We model a network as a directed graph $G(V, E)$, where V is the set of nodes ($|V| = N$) and E the set of directed *edges*. Every edge $(i, j) \in V$ is associated with a set $\mathcal{M}$ of non-negative values, referred to as edge metrics. Thus an edge (i, j) has cost $m(i, j)$ with respect to metric $m \in \mathcal{M}$. Examples of edge metrics are: edge cost (c), bandwidth (b), delay (d), delay jitter (j), loss probability (l).

A path $Pa(s, t)$ is a sequence of vertices $v_s, \ldots, v_i, v_{i+1}, \ldots, v_t$ such that $\forall s \leq i \leq t$, edge $(v_i, v_{i+1}) \in E$.

Upon each metric $m \in \mathcal{M}$, we further define the cost $c_m(s, t)$ of a path $Pa(s, t)$ as a generic function f of the path's edges metrics, or:

$$c_m Pa(s,t) = f(m(v_s, v_{s+1}), \ldots, m(v_i, v_{i+1}), \ldots, m(v_{t-1}, v_t)) \tag{1}$$

Examples of path cost functions are:

$$c_{bw} Pa(s,d) = \min_{(i,j) \in Pa(s,d)} b(i,j) \qquad < bandwidth > \tag{2}$$

$$c_{bf} Pa(s,d) = \min_{(i,j) \in Pa(s,d)} bf(i,j) \qquad < buffer > \tag{3}$$

$$c_h Pa(s,d) = \#edges\ of\ Pa(s,d) \qquad < hop\ count > \tag{4}$$

$$c_d Pa(s,d) = \sum_{(i,j) \in Pa(s,d)} d(i,j) \qquad < delay > \tag{5}$$

$$c_j Pa(s,d) = \sum_{(i,j) \in Pa(s,d)} j(i,j) \qquad < jitter > \tag{6}$$

$$c_l Pa(s,d) = 1 - \prod_{(i,j) \in Pa(s,d)} [1 - l(i,j)] \qquad < packet\ loss > \tag{7}$$

where $b(i, j)$, $bf(i, j)$, $d(i, j)$, $j(i, j)$, $l(i, j)$ are the bandwidth, buffer, delay, delay jitter, and packet loss probability of link (i, j), respectively *.

A QoS routing task consists of finding a path $Pa(s, t)$ suitable to a given application which has end-to-end constraints. These constraints translate into specific path value bounds for one or more of the metrics defined above. According to the definitions above, one can see that bandwidth and buffer are minimum path constraints, while delay and delay jitter are additive constraints, and finally loss is a multiplicative constraint. In particular, we are interested in the "best" path which satisfies a given application's end-to-end constraints, in the sense that such path should use the minimum amount of

*We assume that a QoS application requires a minimum amount of buffering at each router along its path

network resources. Therefore, an optimal path is generally defined as a path with minimum number of hops which still satisfies a given set of metric (upper/lower) bounds.

2.1 Related work

The most general shortest path problem with multiple constraints is known to be NP-Complete (Garey *et al.* 1979). Recent research in the solution of Bellman's Equations with multiattributes (Henig 1994) establishes some conditions on the set of vectors of path values so that an order of preference among the possible paths can be defined, and thus a solution obtained. In general, however, the checking of those conditions on a particular instance of the problem takes exponential time. (Henig 1994) also shows that, for a given class of multiattribute shortest path problems, a polynomial solution is available. Our work addresses a small sub-class of the above mentioned class.

(Wang *et al.* 1995) first modelled the QoS routing problem as a multiple metric shortest path problem. By using standard reduction techniques, they showed that a routing problem subject to any two metrics among cost, delay, loss probability and jitter, is NP-Complete. However, they make no assumptions about the multiple metrics when deriving the impossibility results. Since then, solutions for multiple constrained routing problems have been reduced to either the minimization of a single cost metric and the verification that the path computed satisfies the required bounds (Lee *at al.* 1997); or the definition of a hierarchy of cost functions, to be optimized one at a time (Guerin *et al.* 1997a). Our approach differs from previous research work in that: i) It insists in computing **minimum** hop paths which satisfy multiple metric cost **bounds**; ii) it makes realistic assumptions about delay, loss probability, jitter, and bandwidth metrics within an AS.

3 USING BELLMAN-FORD ALGORITHM WITH MULTIPLE METRICS

Consider the network shown in Figure 1. Arc values represent link delays.

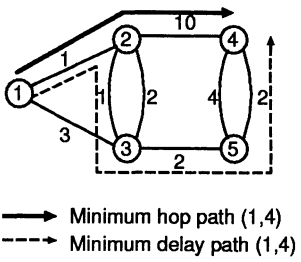


Figure 1 Delay-hop routing example

Let us assume we wish to compute a path between nodes 1 and 4, subject to delay bound, D. We look for the minimum hop path with delay smaller or equal to D. Our choice of paths will obviously depend on the value of D: if $D > 11$, the minimum hop path shown in the figure should be used; if $7 \leq D < 11$, path $(1, 3, 5, 4)$ should be used; if $D = 6$, the minimum delay path shown in the figure is the only alternative. For delay requirements smaller than $D = 6$, no path is available.

The original Bellman-Ford (BF) algorithm was designed to compute shortest paths between a given vertex s and multiple destinations. It did not include multiple metrics. Moreover, path cost function was always considered to be of additive nature, as in the path delay cost above. We wish to extend BF algorithms to handle as many metrics of the types previously defined as possible.

Without loss of generality, we assume vertex 1 as the vertex from which we wish to compute distances to all other network vertices. As usual in BF algorithm, we define D_i^h as the minimum distance with respect to some metric d between vertex 1 and vertex i with at most h number of hops. The Bellman-Ford equation (Cormen *et al.* 1990) is:

$$D_i^{h+1} = \min_{j \in N(i)} [d(i,j) + D_j^h], \qquad \forall i \neq 1 \tag{8}$$

where $N(i)$ is the set of neighbors of vertex i.

Starting with initial conditions - $D_i^0 = \infty$ and $D_1^h = 0$ - BF algorithm iterates eq. (8) until a predefined number of hops H_{max} has been reached, $H_{max} \leq N$ (If the shortest possible path is sought, regardless of number of hops, $H_{max} = N$).

In a regular BF algorithm, it is well known that:

Theorem 1 *(Bertsekas et al. 1992) D_i^h is non-increasing with h, regardless of the cost metric d used.*

Essentially, the non-increasing property of the distance D^h is provided by the **min** operator of eq. (8). Notice also that the search direction defined by the **min** operator is the steepest descend of D^h. Figure 2 illustrates D_i^h dependence on h.

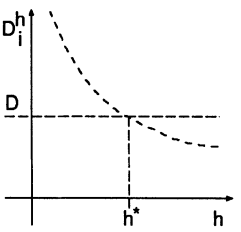


Figure 2 Non-increasing cost function

The first routing problem of interest is:

Problem 1 *Find a minimum-hop route $Pm(s,t)$ with d (delay) cost upper bound $c_d Pm(s,t) \leq D$*

It is not difficult to see that we can use BF algorithm to solve problem 1. More specifically, if h^* is defined to be the shortest number of hops in which $D_i^h \leq D$ *, then:

Theorem 2 *A regular BF algorithm outputs the minimum-hop path with $c_d Pa(s,t) \leq D$ at the first time $D_i^h \leq D$.*

Proof of Theorem 2 *We run BF algorithm until the first time D_i^h falls below D, say at h^*. Since the number of hops always increases as BF progresses, and by assumption h^* is the point in which D_i^h falls below D for the first time, the Theorem is proven (Fig. 2).*

Figure 3 illustrates successive iterations of the Bellman-Ford algorithm for the example shown above. It is worth noting that the Bellman-Ford algorithm has a worst case complexity of $O(N^3)$ (Bertsekas *et al.* 1992), which is higher than the Dijkstra algorithm complexity of $O(N \log N)$. However, the regular Dijkstra does not classify the paths it generates on the basis of hop count, as BF algorithm does. This feature comes handy when dealing with multiple constrained routing problems.

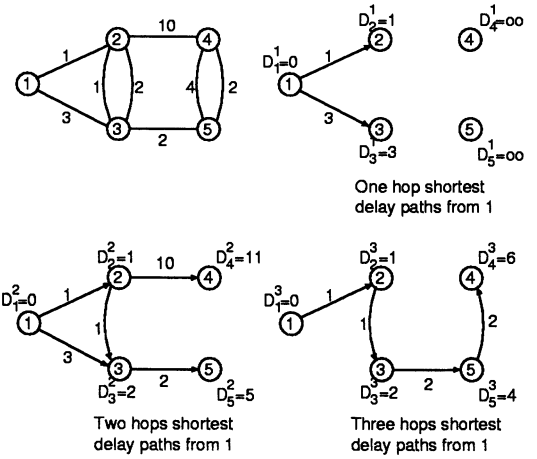


Figure 3 Successive iterations of Bellman-Ford Algorithm

*We will use * to denote optimal paths

We now introduce a second distance M in the BF algorithm, in the following way:

$$M_i^{h+1} = f[m(i,k), M_k^h], \qquad k \; s.t. \min_{j \in N(i)} [d(i,j) + D_j^h] \tag{9}$$

i.e., M_i^h is the path distance with regard to metric m of the path chosen in order to minimize the distance D_i^h. For example, metric m could be any of the metrics defined in Eqs. (2) through (7). Notice that a priori we can not claim any property of M_i^h, as stated in Theorem 1 for D_i^h, unless we assume some correlation between metrics m and h. If such a correlation exists, two cases are of interest: M_i^h is non-increasing with h; M_i^h is non-decreasing with h. Figure 4 illustrates such cases.

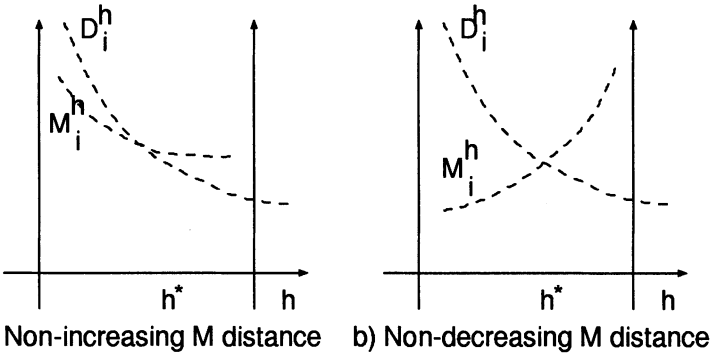


Figure 4 Multiple cost functions

For both cases, we further assume that M_i^h has no discontinuities, namely, any two paths with the same number of hops has the same distance M. So vertical jumps on Figure 4 are forbidden. Given that the path cost functions dealt with are of the form depicted in Fig. 4, one can use the same strategy as before to compute minimum hop paths with bounds on multiple constraints. More specifically, we define the following routing problems:

Problem 2 *Find a minimum-hop route $Pm(s,t)$ with d cost upper bound $c_d Pm(s,t) \leq D$ and m cost upper bound $c_m Pm(s,t) \leq M$.*

Problem 3 *Find a minimum-hop route $Pm(s,t)$ with d cost upper bound $c_d Pm(s,t) \leq D$ and m cost lower bound $c_m Pm(s,t) \geq M$.*

It is not difficult to see that:

Theorem 3 *A regular BF algorithm outputs the minimum-hop path with* $c_d Pa(s,t) \leq D$ *and* $c_m Pm(s,t) \geq M$ *if at the first time the condition* $(D_i^h \leq D)$ *we have* $(M_i^h \geq M)$ *is true for a non-increasing* M^h *distance. If it so happens that* $(M_i^h \leq M)$*, the problem has no solution.*

Proof of Theorem 3 *Follows from the non-increasing property of distance* M^h *and Theorem 2.*

Theorem 4 *A regular BF algorithm outputs the minimum-hop path with* $c_d Pa(s,t) \leq D$ *and* $c_m Pm(s,t) \leq M$ *if at the first time the condition* $(D_i^h \leq D)$ *we have* $(M_i^h \leq M)$ *is true for a non-decreasing* M^h *distance. If it so happens that* $(M_i^h \geq M)$*, the problem has no solution.*

Proof of Theorem 4 *Follows from the non-decreasing property of distance* M^h *and Theorem 2.*

Figure 5 illustrates two metric constrained routing problems and their solutions.

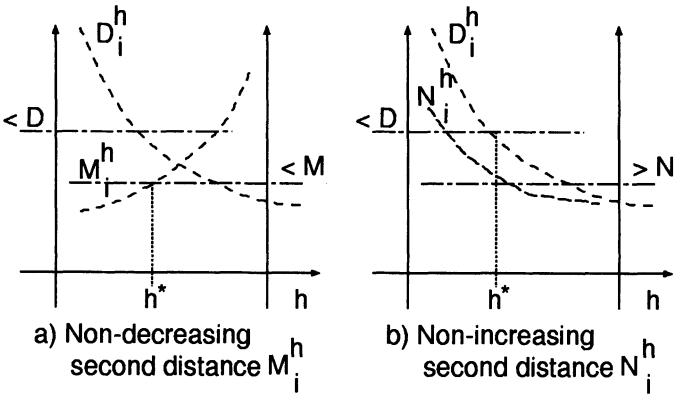


Figure 5 Two constrained BF solvable routing problems

Notice that if we are seeking an upper bound for a non-increasing distance N_i^h, or a lower bound for a non-decreasing distance M_i^h, the doubly constrained minimum hop problem can not be solved by the Bellman-Ford algorithm. Also, the monotonic non decreasing/increasing properties of the distances M_i^h and N_i^h are key to claim that the problem has no solution if the two bounds are not met. For instance, lets assume two distances D_i^h and M_i^h, based on generic metrics d and m, whose dependency with the number of hops is displayed in Figure 6.

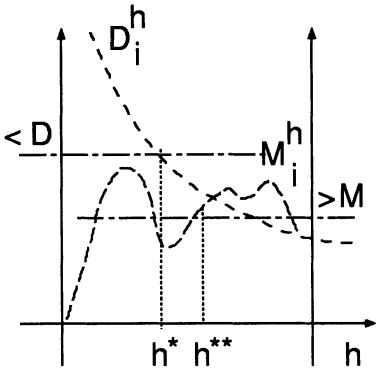


Figure 6 When BF-algorithm fails

The path represented by h^* satisfies the upper bound D, although it violates the lower bound M. However, according to the figure, it is possible to continue lowering the distance D^h and find a path with $h^{**} > h^*$ satisfying the lower bound M. Notice that we can not claim that the path determined by h^{**} is the shortest one (in number of hops) sought satisfying both bounds. This is so because, once the upper bound D is satisfied, a search on the steepest descend direction on distance D_i^h might no longer be the best search strategy towards finding a minimum hop path respecting the bound M. Conversely, although a search on the steepest descend direction on distance M_i^h will lead to a shortest path on h satisfying the lower bound M, this search strategy is not guaranteed to yield a monotonic decrease on the distance D_i^h (see the definition of distances D_i^h and M_i^h).

Now for the main theorem. For a multiple constrained routing problem, define distance D_i^h as in eq. (8) wrt a given metric $d(i,j)$. Then:

Theorem 5 *Partition the multiple constrained minimum hop routing problem into subproblems by pairwising metric d with each other metric. The original multiple constrained problem can be solved by a multiple metric BF algorithm if and only if each subproblem is of the form defined by either Theorems 3 or 4.*

Proof of Theorem 5 *If follows from Theorems 3 and 4 and the fact that multiple metrics interact only through the distance D_i^h which defines the search direction (see eq. 9).*

One more observation is due. For a generic metric, the verification whether a given distance has non-increasing/non-decreasing property might take exponential time (see (Henig 1994) for further discussions on this issue). However, this property is called upon only when the multiple metric BF algorithm fails to find a path, to claim that the problem has no solution. If a path satisfying

the multiple constraints is found, Theorem 1 implies that this path is indeed the shortest hop path sought.

4 SOLVING ROUTING PROBLEMS WITH MULTIPLE CONSTRAINTS

We now take a closer look at practical metrics defined in section 2, and their dependency on the number of hops in a typical network scenario. Although in establishing the theory we did not assume any particular scheduling discipline exercised at the routers, for realism we draw our examples from recent research work in the provision of delay and delay jitter bounds.

Bandwidth

Typically, bandwidth availability is a metric in which we should not expect any correlation with hop count. This is because bandwidth availability in a link depends not only on the distribution of flows but also on the routing strategy utilised to route flows across the network. Figures 7 a) and b) illustrate a network with available bandwidth and path bandwidth dependency on the number of hops, respectively.

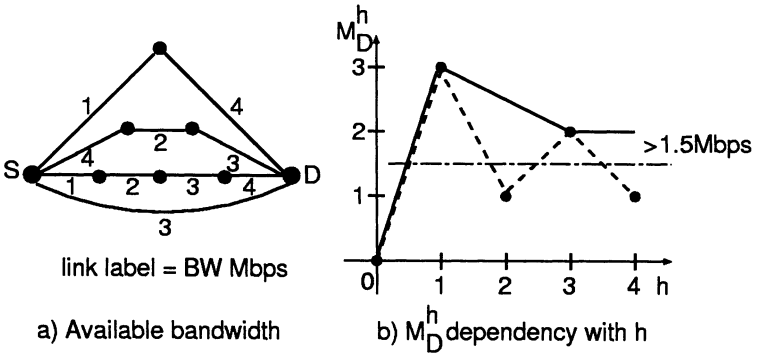


Figure 7 Typical path bandwidth dependency with number of hops

The dashed line shows a typical behavior of the distance M_D^h. In a search for a QoS path requiring minimum bandwidth, say $1.5Mbps$, there is no need in considering paths that fall below a $1.5Mbps$ threshold line, as shown in the figure. By disregarding all links in the graph with bandwidth less than the required, we reduce our search space to paths with bandwidth at least as much as the requested. This is equivalent to redefining distance D_i^h as:

$$D_i^{h+1} = \min_{j \in N(i) \& m(i,j) > M} [d(i,j) + D_j^h], \qquad \forall i \neq 1 \tag{10}$$

The solid line in figure 7 b) shows the new distance M_D^h dependency when links with $1Mbps$ are not considered in the routing problem. Notice that if we were searching for a path whose QoS requirements translates into minimum bandwidth only, the path in which M_D^h crosses the threshold line for the first time should be used. Notice also that the curves assume $M_D^0 = 0$ as the initial condition for the routing algorithm.

Buffer

In general, the amount of buffers allocated to a given session is expected to be related to the bounds on loss probability and delay guaranteed by the router. Buffer requirements per node, thus, depend on the scheduling and service disciplines exercised by the router (see (Zhang 1995) for a survey on relevant service disciplines for deterministic delay bounds). If a given session needs a specific amount of buffers for reasons other than loss bounds (e.g., delay bounds provided by some scheduling discipline (Zhang 1995)), we can use definition 3 for buffer path cost. According to definition 3, it is easy to see that a distance B_i^h has the same dependance on the number of hops as for the bandwidth case described above. Thus, for a given QoS connection with buffer requirement $> B$, a good strategy is to eliminate all links without the minimum buffer storage.

The path cost definition for buffers assumes a uniform buffer requirement along all routers of a given path. Recent scheduling disciplines (Georgiadis *et al.* 1996) imply buffer allocation which grows linearly with the hop count. In this case, we can also redefine distance D_i^h so as to reduce the number of metrics in the routing problem, as:

$$D_i^{h+1} = \min_{j \in N(i) \& bf(i,j) > h \times B} [d(i,j) + D_j^h], \qquad \forall i \neq 1 \tag{11}$$

Notice that equation (11) is well defined since h is known at all times of the BF algorithm computation.

Loss probability

We believe that, at least within an autonomous system (AS), the various routers will typically advertise a constant loss probability for a given class of traffic. This being the case, we should expect that a distance L_D^h dependency with hop count as exemplified in Figure 8.

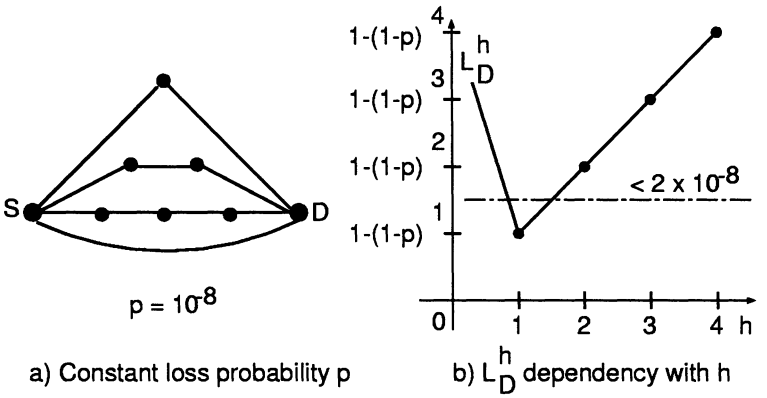


Figure 8 Typical path loss dependency with number of hops

We require $L_D^0 = \infty$ if there is no zero hop path for the source destination pair S, D. Notice that as soon as $L_D^h \neq \infty$, the behavior of L_D^h is monotonically increasing (almost linear) with the number of hops. Figure 8 b) shows a QoS loss requirement of $L < 2 \times 10^{-8}$. If this is the only QoS requirement, the one hop path in the figure is the answer to our QoS routing problem.

Delay Jitter

We anticipate that, as in the case of loss probability, all routers of a given AS will advertise the same delay jitter bounds. This is essentially because propagation delays are assumed constant within an AS, which makes delay jitter dependent on the scheduling and service disciplines exercised by the router. For deterministic jitter bounds, the same values are expected to be advertised for a given traffic class (Georgiadis *et al.* 1996) (Zhang 1995). In this case, the dependency of the distance J_D^h with the number of hops is linear, thus similar to the distance L_D^h shown above.

Delay

With regard to hop delays, a delay metric $d(i,j)$ will have various values for different links. Since $d(i,j)$ includes link propagation delays plus queueing delay and service time (packet size), even if queue waiting times are similar across routers within an AS, propagation delays and service times vary wildly. Therefore, it is not legitimate to assume any particular dependency behavior of D_i^h with the number of hops. However, if we define the distance D_i^h as in eq. (8), Theorem 1 guarantees the non-increasing property of D_i^h necessary to compute in polynomial time QoS paths with multiple constraints, regardless of the $d(i,j)$ metric behavior.

From the characteristics of the various metrics of interest and from Theorem 5, it is easy to check that a multiple constrained routing problem constructed with any subset of the metrics discussed above is solvable by a multiple metric

BF algorithm. Notice that the restrictive uniform behavior on loss probability and delay jitter metrics assumed is not strictly necessary for the QoS aware Bellman-Ford algorithm to work, but only that their corresponding distances L_i^h and J_i^h be non-decreasing functions with h. For instance, in Figure 8, if the link of the one hop path has loss probability p_1, the links of the two hop path have loss probability p_2, and so on, a QoS aware Bellman-Ford will work provided that: $1-(1-p_1) \leq 1-(1-p_2)^2 \leq 1-(1-p_3)^3 \leq 1-(1-p_4)^4$. This means that the set of routing problems solvable by the multiple metric Bellman-Ford algorithm defined in the previous section is larger than the set considered here. We have considered uniform link values for these metrics because we can easily check the non-decreasing distances property with the number of hops. For a generic metric, however, this checking takes exponential time (see (Henig 1994) for further discussions on this issue). For generic loss probability and delay jitter metrics, one can still use the multiple metric Bellman-Ford algorithm defined in the previous section. Strictly speaking, the only limitation is that, in case a feasible solution is not found, one can not be sure that indeed such a feasible path does not exist.

5 SIMULATION STUDY

In this section, we illustrate the advantages of our QoS routing approach, namely, minimizing hop distance while satisfying various metric bounds. Our figure of merit is session acceptance probability, which is evaluated for various network loads. We assume each session has specified QoS bounds on the metrics considered, possibly conveyed by RSVP signalling upon entering the Autonomous System (AS) cloud (Guerin *et al.* 1997b). Although a session may traverse several ASs, where each AS requires a QoS routing solution of its own, we consider a single AS only. Therefore, our bounds do not represent end-to-end requirements, but rather incremental budgets to be fulfilled within each intermediate AS on an end-to-end session traversing many Autonomous Systems.

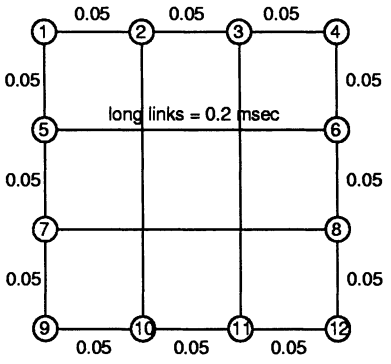
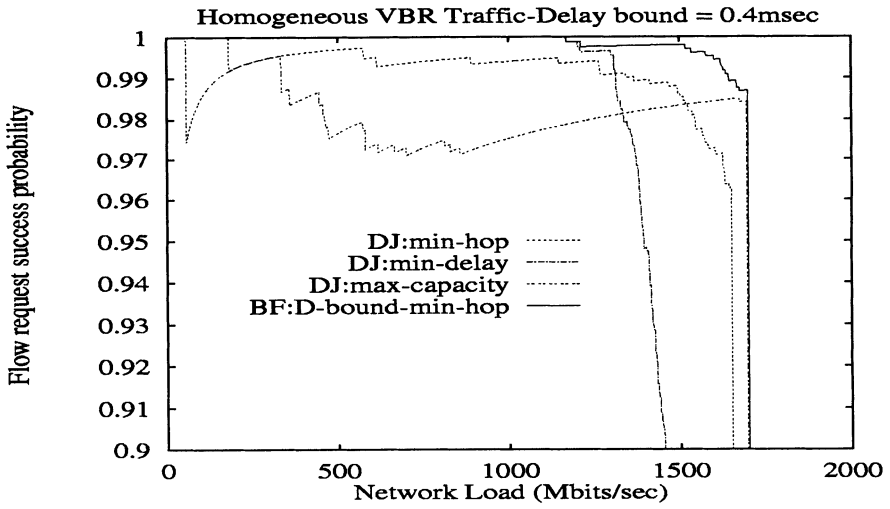


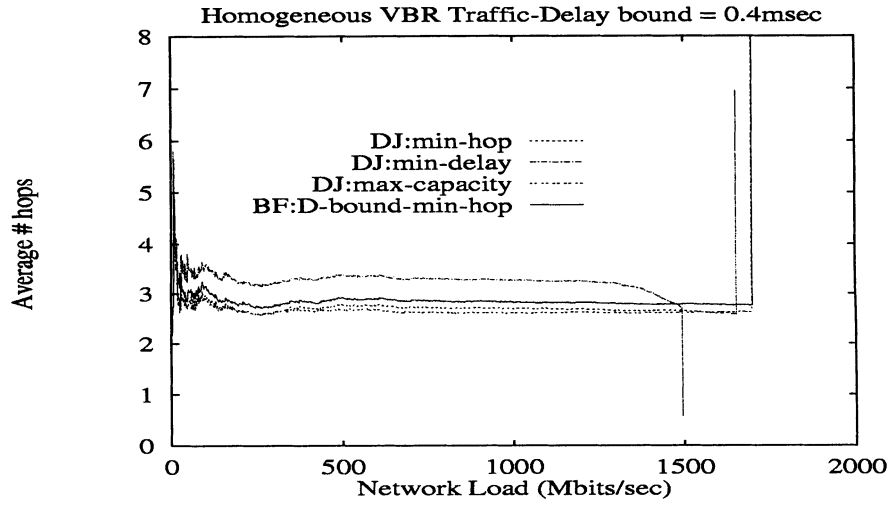
Figure 9 Small Autonomous System

We assume a small network of 12 nodes, depicted in Figure 9. The network has a very simple topology, so that we can force the various routing algorithms to compute different routes. The network diameter is four. Links have 155Mbps speed and propagation delays are as labelled (msec) in the figure. Propagation delays were chosen to reflect a 5Km island. No queueing nor scheduling delays are explicitly considered in the simulations, as they could be easily added to the propagation delays. Buffers are provided on per output basis. A flooding scheme is used to convey link state information across the network. Link state messages are generated periodically, approximately at every 50msec. At each network node, new reservation requests arrive according to a Poisson process, with an average rate of 200 requests per second per node. Once resources are reserved within the network, they are held for the entire duration of the simulation experiment. Thus, eventually the network becomes saturated. The figure of merit is the number of requests which are accepted, that is, the connection success rate.

In the first simulation study, we consider delay bounds only. The sessions require bandwidth $\geq$ 1.5 Mbps, and delay $\leq$ 0.4 msec. Four routing strategies are considered: 1) Dijkstra minimum hop algorithm: a minimum hop path is first generated. If the path satisfies the delay bound, a reservation message is sent through the new path, otherwise, the session request is rejected; 2) Dijkstra minimum delay algorithm: a minimum delay path is computed. The same check on delay bounds is performed; 3) Dijkstra widest capacity algorithm: the path with maximum available capacity is computed. The resulting path is checked against the delay bound, as before; 4) Bellman-Ford delay-constrained minimum hop algorithm: a minimum hop path among all paths within the delay budget is computed. All routing strategies eliminate at each session request the links with insufficient bandwidth. The results are shown in Figure 10.



a) Request success rate



b) Path average # of hops

Figure 10 Performace of various QoS routing strategies for delay sensitive flows

The Dijkstra minimum hop strategy is quite successful in loading the network up to a high saturation point. Some requests, however, are rejected even

for light network loads. This is mainly because the min-hop strategy is not QoS aware; some of the paths generated by min-hop routing violate the delay bound. For instance, a path request $(9,4)$ will likely generate the path $(9,7,5,6,4)$, which has delay $= 0.45msec > 0.4msec$, whereas the longer hop path $(9,7,5,1,2,3,4)$ has delay $= 0.3$, thus within the delay bound.

A Dijkstra minimum delay strategy, on the other hand, provides a very high session acceptance rate for light to medium network load states, since if a path is found, this path is guaranteed to comply with the delay bound. However, as some paths are unnecessarily long in hop count (high average hop count, see Fig. 10 b)), the load saturation point is the worst among all routing strategies surveyed. For instance, a request $(1,4)$ will likely result in the path $(1,2,3,4,6)$, with delay $= 0.2$ msec, whereas the less wasteful path (in network resources) $(1,5,6)$ is available.

Interestingly enough, the Dijkstra widest capacity strategy leads to the best network utilization among all routing strategies. However, it also has the worst performance in success rate, because as the network load increases, uncongested paths quickly become too long to support the delay bound required. For instance, a request $(10,9)$ will likely generate path $(10,2,1,5,7,9)$, which violates the delay bound required, if the delay compliant path $(10,9)$ has less but sufficient bandwidth available to carry the session.

Lastly, Bellman-Ford delay bound minimum hop strategy has the best performance both regarding session request success rate and maximum network load. This is because it selects paths with minimum number of hops within the delay budget.

The average number of hops depicted in Fig. 10 b) agrees with our observations. Notice that, when the network is heavily saturated, all but the minimum delay routing algorithm manage to still find very long (in hop count) suitable paths. That is because the minimum delay routing strategy has used all long paths before the network got into a heavy load state.

The Bellman-Ford QoS aware routing algorithm tries to choose the minimum hop path, among those which satisfy the QoS constraints. We believe that this strategy is more prone to errors than others if the algorithm uses stale information, due to network latency. We have confirmed this conjecture by repeating the simulation above, with a broadcast frequency 100 times slower than before. We have tracked the number of paths believed to be QoS compliant, but which failed to get the resources reserved, due to inaccurate link state information. Since the load/request success rate profile remains unaltered, we report only the percentage of unsuccessful requests due to stale link information (Fig. 11).

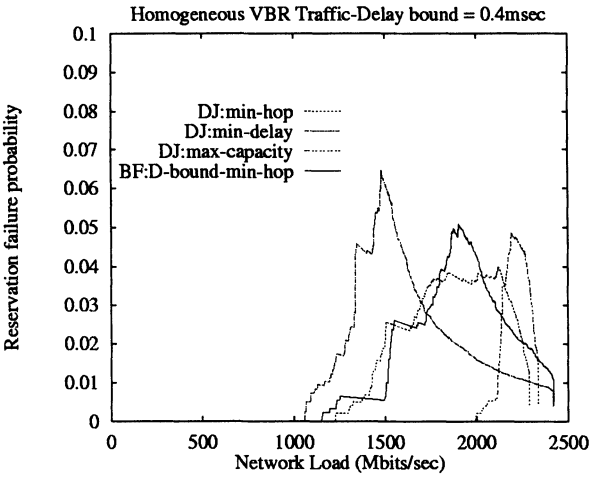
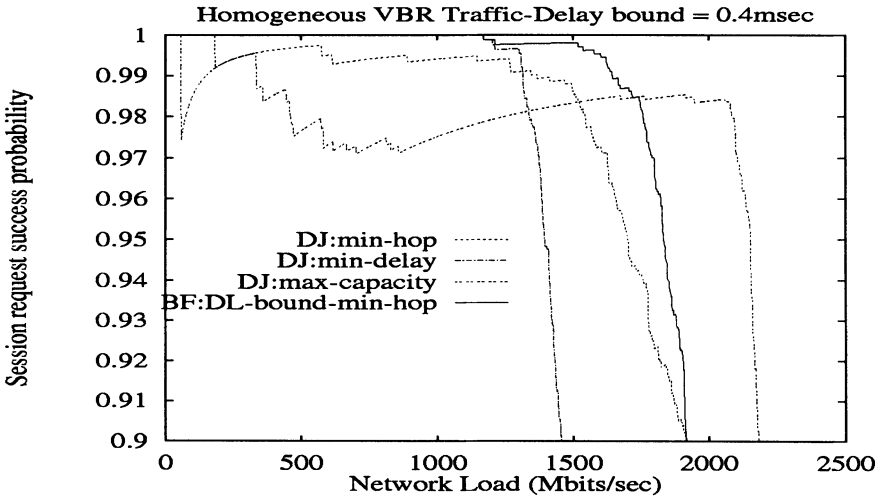


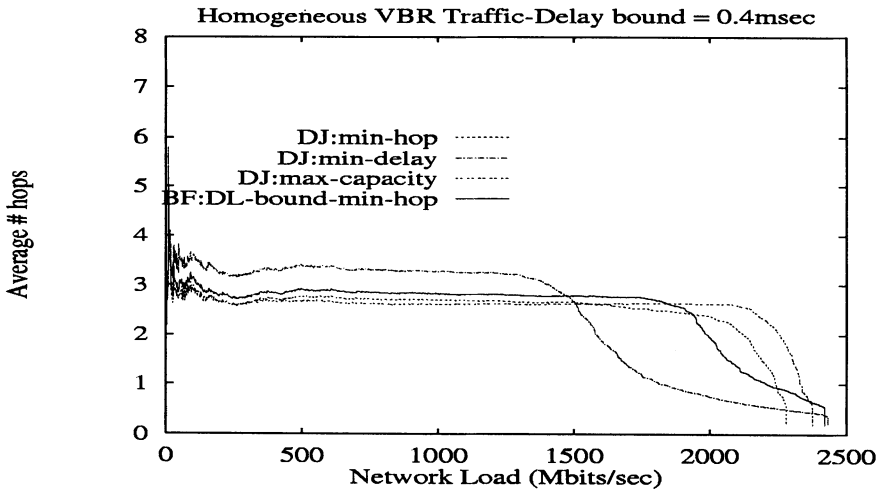
Figure 11 Impact of stale link state information on the various routing strategies

In the next experiment, we have included an additional constraint in our problem, namely, the loss probability metric. Each router must provide a packet loss guarantee of 10^{-8} for each of its links, and the QoS application requires a loss not superior to 3×10^{-8} in addition to an end to end delay $\leq 0.4msec$. Results are depicted in Figure 12.

First note that the network saturation point is higher than previously for all algorithms. This is explained by the fact that the loss probability constraint tends to block long sessions (in hop count), which improves network utilization. The additional observation to be made by inspection of Fig. 12 is that the Dijkstra widest capacity strategy leads to the best network utilization (at saturation) among all routing strategies, even better than the Bellman-Ford QoS routing strategy. However, max-cap gives an unnacceptably high rejection rate in the non-saturated region. We argue that this is because again the loss probability constraint limits the feasible sessions to a very small number of hops. Within this scenario, the Dijkstra widest capacity routing strategy is successful in routing only very short (in hop count) sessions. Since the QoS Bellman-Ford is able to establish longer sessions, the maximum network load is expected to be smaller than the Dijkstra widest capacity. This can be confirmed by the average number of hop curve displayed by Fig. 12 b). Notice also that the high average hop count of Dijkstra minimum delay strategy is again wasteful, leading to the worst network saturation point. Finally, since long paths (in hop count) are no longer suitable due to the loss probability constraint, no spike is present at the left side of the average hop count curves, as it was the case with the previous simulation experiment.



a) Request success rate



b) Path average # of hops

Figure 12 Performace of various QoS routing strategies for delay and loss sensitive sessions

6 CONCLUSIONS AND FUTURE WORK

We have studied a class of QoS routing which can be solved with a Bellman-Ford algorithm, when the main objective fuction is the number of hops. We have shown that, for delay jitter and loss probability metrics with certain properties (e.g., constant among links of an AS), a BF algorithm is capable of solving a minimum hop, delay, delay jitter, loss, and bandwidth constrained routing problem. The class of multiple metrics considered in this document is still a bit restrictive. (Henig 1994), among others, has shown that a large class of multiattribute shortest path problems can be solved efficiently. The question whether this large class includes any routing problem of interest for QoS applications in multimedia high speed packet networks is an important research topic.

We have compared a QoS Bellman-Ford algorithm with a widest shortest path, and a pure minimum single metric strategy, including delay and hop count. We have exemplified tradeoffs in using these various routing strategies in reserving resources within an Autonomous System.

Although we have considered a centralized version of the Bellman-Ford algorithm, a distributed version is also possible, if QoS routing research moves towards distributed algorithms. However, we have detected a recent trend towards centralized algorithms, such as QOSPF, rather than distributed ones. Since centralized algorithms require large databases for topology information, QoS hierarchical routing, where routing information is aggregated, is an important research direction. Therefore, we are currently extending the present study to QoS routing across ASs, in a hierarchical routing fashion. The present work points to interesting directions regarding the budget of end-to-end QoS bounds among the various Autonomous Systems. We are also studying how aggregation of resource information may affect the routing algorithms studied in this document.

REFERENCES

D. Bertsekas and R. Gallager (1992) Data Networks, *Prentice-Hall.*

Cormen, Leiserson and Rivest (1990) Introduction to Algorithms, *McGraw-Hill.*

M. R. Garey and D. S. Johnson (1979) Computers and Intractability, *Freeman*, San Francisco.

L. Georgiadis, R. Guerin, V. Peris (1996) Efficient Network QoS Provisioning Based on per Node Traffic Shaping, *In Proceedings of IEEE INFOCOM96*, vol.1, pp. 102-110.

R. Guerin, A. Orda, and D. Williams (1997a) QoS Routing Mechanisms and OSPF Extensions, *In Proceedings of IEEE GLOBECOM97*, Vol. 3, pp. 1903-1908, Phoenix, Arizona.

R. Guerin, S. Kamat, and S. Herzog (1997b) QoS Path Management with

RSVP, *In Proceedings of IEEE GLOBECOM97*, Vol. 3, pp. 1914-1918, Phoenix, Arizona.

M. Henig (1994) Efficient Interactive Method for a Class of Multiattribute Shortest Path Problems, *In Proceedings of Management Science*, Vol. 40, # 7, pp. 891-897.

K. Lee, K. Kim, H. Choi, K. Kim, and S. Kim (1997) QoS Based Routing for Integrated Multimedia Services, *In Proceedings of GLOBECOM97*, Vol.2, pp. 1047-1051, Phoenix, Arizona.

C. Pornavalai, G. Chakraborty, and N. Shiratori (1997) QoS Based Routing Algorithm in Integrated Services Packet Networks, *In Proceedings of International Conference on Network Protocols*, Atlanta, Georgia, pp. 167-175.

Z. Wang and J. Crowcroft (1995) Bandwidth-Delay Based Routing Algorithms, *In Proceedings of GLOBECOM95*, Vol.3, pp. 2129-2133.

H. Zhang (1995) Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks, *In Proceedings of the IEEE*, 83(10), pp. 1374-1396.

W. Zhao and S. K. Tripathi (1997) Routing Guaranteed Quality of Service Connections in Integrated Services Packet Networks, *In Proceedings of International Conference on Network Protocols*, Atlanta, Georgia, pp. 175-182.

7 ACKNOWLEDGEMENTS

This work has been partially supported by CNPq under Grant 201597/93-4(NV) and by GTE.

8 BIOGRAPHY

Dirceu Cavendish was born in Recife, Brazil. He received his bachelor degree in Electronics from Federal University of Pernambuco, Brazil in 1986. He spent five years working with telecommunications in the Business Communications Division of Philips, as a Development Engineer. He received his M. S. in Computer Science from Kyushu Institute of Technology, Japan, in 1994. He is currently a Ph. D. candidate at Computer Science Department-UCLA, working with congestion control and routing in Quality of Service supportive high speed networks.

Mario Gerla was born in Milan, Italy. He received a graduate degree in engineering from the Politecnico di Milano, in 1966, and the M.S. and Ph.D. degrees in engineering from UCLA in 1970 and 1973, respectively. He joined the Faculty of the UCLA Computer Science Department in 1977. His research interests cover the performance systems and high speed computer networks (B-ISDN and Optical Networks).