QoS routing in packet switched networks - novel algorithms for routing with incomplete information

J. Levendovszky[1], G. Retvari[2], T. David[1], A. Fancsali[1], Cs. Vegso[1]
[1]High Speed Networks Laboratory, Department of Telecommunications,
[2]High Speed Networks Laboratory, Department of Telecommunications and Telematics
Budapest University of Technology and Economics,
e-mail: levendov@hit.bme.hu Pazmany P. setany 1/D., Budapest 1117, Hungary

Abstract

This paper investigates QoS routing in IP networks. The major concern is to select paths to fulfill end-to-end delay or minimum bandwidth requirements. Novel algorithms are developed to tackle routing with incomplete information, when link measures are subject to random fluctuations described by some given p.d.f.-s. The new algorithms are based on either assuming Gaussian link delay distribution or using large deviation theory to find the most likely path. The proposed methods are capable of QoS routing in polynomial time.
Keywords: QoS routing, incomplete information, Bellman Ford algorithm, Chernoff bound.

1 Introduction

One of the major challenges in IP networking is to ensure QoS routing, which selects paths to fulfill end-to-end delay or bandwidth requirements [1, 2, 11] as opposed to traditional shortest path routing. Because of the manifold optimization criteria, QoS routing can prove to be much harder than the problem of finding optimal paths based on merely the hop count [14]. OSPF and PNNI routing protocols offer two (or many) levels, where routing can be performed in a hierarchical manner [13]. Subnetworks on a given level of the hierarchy are abstracted as "nodes" for a higher layer and delay information in those subnetworks are aggregated into an average QoS parameter. On the other hand, randomly fluctuating traffic load on links can also result in random delays. Thus link delays are periodically advertised when the delay surpasses a given threshold (e.g. in PNNI and QOSPF standards, see [10, 13]). These thresholds are defined in advance. This prompts us to take delays into account as random variables characterized by their probability distribution functions over the interval between two reported thresholds [4, 15]. The distribution of these thresholds (and therefore the length of the intervals over which the link delay is not fully characterized) can be equidistant or non-equidistant. In practice non-equidistant thresholds are used, since in this case the impact on network utilization by sending only signaling information (part of the available bandwidth is used for carrying information about link delays) is minimized.

The phenomena described above give rise to the task of routing with incomplete information. Namely, how to select paths to fulfill end-to-end delay requirements in the lack of the exact values of link delays [4, 15, 16]. Routing is then perceived as an optimization problem to search over different quality paths, where the quality of a path is determined by the probability of meeting the end-to-end QoS requirement [3, 4]. Unfortunately, routing with incomplete information in general cannot be reduced to the well-known Shortest Path Routing (SPR), thus it cannot be solved in polynomial time.

In this paper, novel solutions will be presented to the problem of QoS routing with incomplete information. These new methods are capable of finding an optimal path in polynomial time.

The methods are tested by extensive simulations and the results are discussed in the following structure:

- In *Section 2* the underlying model is introduced and a brief summary of notations and some initial results are given.
- In *Section 3* new QoS routing algorithms are proposed based on either assuming link delays subject to Gaussian distribution or using large deviation theory. The polynomial complexities of these algorithms are proven.
- In *Section 4* a detailed numerical analysis is given where the performances of the algorithms are tested by extensive simulations.

2 Routing with incomplete information - the model

To model the routing problem let us assume that the following quantities are given:

- there is graph $G(V, E)$ representing the network topology, where nodes are denoted by index $u \in V$ and links referred to as node pairs $(u, v) \in E$;
- each link $(u, v) \in E$ has a QoS descriptor $\delta_{(u,v)}$ which is assumed to be a random variable subject to a probability distribution function $F_{(u,v)}(t) = P\left((\delta_{(u,v)} < t\right);$
- we assume link independence, i.e. the link delays are supposed to be independent random variables;
- random variable $\delta_{(u,v)}$ is referred to as "bottleneck measure" if the smallest link measure contained in the path determine the quality of the path ($\delta_{(u,v)}$ is a "bandwidth-like" variable);
- random variable $\delta_{(u,v)}$ is referred to as "additive measure" if the sum of link measures contained in the path determine the quality of the path ($\delta_{(u,v)}$ is a "delay-like" variable);
- there is an end-to-end QoS criterion (e.g. $\min_{(u,v)\in R} \delta_{(u,v)} \geq W$ for some W in the case of bandwidth requirement or $\sum_{(u,v)\in R} \delta_{(u,v)} < T$ for some T in the case of end-to-end delay requirement).

The objective is to find an optimal path $\tilde{R}$ which most likely fulfills the given QoS criterion, namely:

$$\tilde{R}: \max_{R} P\left(\min_{(u,v)\in R} \delta_{(u,v)} \geq W\right) \tag{1}$$

in the case of a bottleneck measure or

$$\tilde{R}: \max_{R} P\left(\sum_{(u,v)\in R} \delta_{(u,v)} < T\right) \tag{2}$$

in the case of an additive measure.

The path $\tilde{R}$, introduced above, will be referred to as the Most Likely Path (MLP). In the case of additive routing we will refer to this problem as Additive Routing with Incomplete Information (ARII).

It is well known that Shortest Path Routing (SPR) can be solved in polynomial complexity by Dijsktra or Bellman-Ford algorithms. Therefore, mapping an MLP problem into an SPR is equivalent with proving that MLP can be solved in polynomial time. The corresponding link measure on which basis the SPR algorithm selects the shortest path is, in general, denoted by $\kappa_(u,v),\ (u,v) \in E$. As a result, our aim throughout the paper is to prove that under certain circumstances the search for MLP can be done by an SPR algorithm and to find the corresponding κ.

The following lemma establishes that a bottleneck measure MLP can easily be solved by using traditional SPR algorithms.

Lemma 1:
The solution of $\tilde{R}:\ \max_R P\left(\min_{(u,v)\in R} \delta_{(u,v)} \geq W\right)$ is equivalent to solving a traditional shortest path problem with the metric assigned to link (u,v) being $\kappa_{(u,v)} = -log\, P\left(\delta_{(u,v)} > W\right)$.

Proof:
We seek the path

$$\tilde{R}:\ \max_R P\left(\min_{(u,v)\in R} \delta_{(u,v)} \geq W\right),$$

which is equivalent to

$$\tilde{R}:\ \max_R P\left(\bigcap_{(u,v)\in R} \delta_{(u,v)} \geq W\right).$$

Since $P\left(\bigcap_{(u,v)\in R} \delta_{(u,v)} \geq W\right) = \prod_{(u,v)\in R} P\left(\delta_{(u,v)} \geq W\right)$, one can write

$$\tilde{R}:\ \min_R \sum_{(u,v)\in R} -\log P\left(\delta_{(u,v)} \geq W\right).$$

Therefore assigning measure κ to link (u,v) as $\kappa_{(u,v)} := -\log P\left(\delta_{(u,v)} \geq W\right)$ MLP routing can indeed be solved by SPR.
Q.E.D.

Based on this lemma, seeking an MLP with respect to bandwidth requirement becomes a relatively easy task.
If the link descriptor is delay, then QoS routing yields an intractable problem stated by the following lemma:

Lemma 2 (Guerin at al):
The find a solution to ARII $\tilde{R}:\ \max_R P\left(\sum_{(u,v)\in R} d_{(u,v)} < T\right)$ in general is NP hard.

The proof is based on the fact that the problem of $\tilde{R}:\ P\left(\sum_{(u,v)\in R} d_{(u,v)} < T\right) > \pi$ (where $0 < \pi < 1$ is some given threshold) is NP hard. For further details, see [4].

One way to make ARII tractable is to reduce it to the bandwidth problem. This gives rise to "rate based" routing algorithms [4]. In practice, under certain scheduling scenarios (such as Weighted Fair Queuing Scheduler, Rate Controlled Earliest Deadline First schedulers), end-to-end delay on route R can be approximated as

$$d(R) = \frac{\alpha_n}{w} + \sum_{(u,v)\in R} r_{(u,v)}. \tag{3}$$

Here α_n depends on the flow's burst and the maximal packet size, whereas $r_{(u,v)}$ is a static link propagation delay and w is the minimal bandwidth which can be guaranteed to the flow. Thus,

in this case ARII reduces to the following problem of finding $\tilde{R}:\ \max_R P(d(R) < T)$.
Unfortunately, this problem is still NP hard as shown in [4]. But assuming that $r_{(u,v)}$ can take their values from a discrete set and the link descriptor is the available bandwidth and by setting $r = \sum_{(u,v)\in R} r_{(u,v)}$, the problem can still be solved by SPR, as was shown in [4]. In this way, rate based routing can still be performed in polynomial time.
Unfortunately, rate based bounds are rather crude, thus the method above yields only suboptimal solutions.

3 Novel algorithms for additive routing with incomplete information

In this section we demonstrate that ARII can be solved in polynomial time under certain assumptions. Let us suppose that links in each time instant (this time instant is set up by the network operator) advertise their delay to the nodes, in the following fashion:

- The delay axis (the set of possible values of the link delays) is covered with a grid $\mathcal{A} = \{t_i,\, i = 1, ..., Z\}$.
- At each time instant link (u, v) advertises the last t_i value, its link delay have exceeded, which implies that $\delta_{(u,v)} \in (t_i, t_{i+1})$.

On these premises we can derive new algorithms which can find an MLP in polynomial time.

3.1 Solving ARII under Gaussian hypothesis

Based on the description above, we assume that the concrete delay is unknown in the interval $\delta_{(u,v)} \in (t_i, t_{i+1})$, after advertising that the link delay has surpassed level t_i,. This prompts us to regard $\delta_{(u,v)}$ as a Gaussian random variable, which has normal distribution over the interval (t_i, t_{i+1}). Despite the fact that a Gaussian distribution is defined over an infinitely long interval, one can fit a Gaussian distribution $\Phi(m, \tilde{\sigma})$ over a finite interval with an ϵ error by solving the following approximation task:

$$m := \frac{t_{i+1} + t_i}{2} \quad \text{and} \quad \tilde{\sigma} : \int_{t_i}^{t_{i+1}} \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(x-m)^2}{2\sigma^2}} dx = 1 - \epsilon \tag{4}$$

Furthermore, let us tentatively assume that grid $\mathcal{A}$ is equidistant, meaning that $\mid t_{i+1} - t_i \mid$ is constant for all i. ARII can then be solved by SPR in polynomial time according the following theorem:

Theorem 1:
If grid $\mathcal{A}$ is equidistant and one defines parameters $m_{(u,v)} = (t_i+t_{i+1})/2$ and $\tilde{\sigma} : \int_{t_i}^{t_{i+1}} \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(x-m)^2}{2\sigma^2}} dx = 1 - \epsilon$ then ARII reduces to an SPR with the link metric $\kappa_{(u,v)} = m_{(u,v)}$. Therefore, QoS routing with incomplete information can be performed by the BF algorithm with $O(\mid V \mid^3)$ complexity.

Proof:
The uncertainty of delay on link l, $\delta_{(u,v)} \in (t_i, t_{i+1})$, is captured by a Gaussian distribution with expected value $m_{(u,v)}$ and variance σ defined by solving the approximation problem in (4). Note that the variance is uniform on each link, owing to the equidistant nature of the grid. Since $\delta_{(u,v)}$ is assumed to be a Gaussian variable, the overall path delay $d(R) = \sum_{(u,v)\in R} d_{(u,v)}$ is also Gaussian with expected value $m = \sum_{(u,v)\in R} m_{(u,v)}$ and with variance $\sqrt{n}\sigma$ (in the case of an

n-hop path). This is due to the fact that link delays are assumed to be independent random variables and the convolution of Gaussian distributions is also Gaussian. Then

$$P(d(R) < T) = \Phi\left(\frac{D - \sum_{(u,v)\in R} m_{(u,v)}}{\sqrt(n)\sigma}\right), \tag{5}$$

where $\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-u^2/2} du$ is the standard Gaussian distribution function. From this it is clear that

$$\tilde{R}: \max_R P\left(\sum_{(u,v)\in R} \delta_{(u,v)} < T\right)$$

reduces to

$$\tilde{R}: \min_R \sum_{(u,v)\in R} m_{(u,v)}$$

which is an SPR with metric $m_{(u,v)}$ assigned to link (u,v).
One must note that n in the denominator in expression (5) denotes the number of hops of the path. Therefore, this algorithm is only of use in the case of n hop routing. But performing the corresponding BF algorithm for each possible hop number, one can obtain $\tilde{R}(n)$ for each $n = 1, ..., \mid V \mid -1$. Then the probability

$$P\left(\sum_{(u,v)\in \tilde{R}(n)} \delta_{(u,v)} < T\right) = \Phi\left(\frac{D - \sum_{(u,v)\in \tilde{R}(n)} m_{(u,v)}}{\sqrt(n)\sigma}\right)$$

can be compared and the best path can be selected. Thus the overall complexity is $O(\mid V \mid^3)$ which is due to the quadratic complexity of BF algorithm plus the n hop routing for each $n = 1, ..., \mid V \mid -1$.
Q.E.D.

Unfortunately, the result above only holds for the case of an equidistant grid where link delays are advertised over uniform intervals. In practice, however, the grid is not equidistant (in the case of larger link delays the thresholds are distributed in an exponential manner, meaning the larger the link delay the less information is available on its value as the corresponding p.d.f. has larger support). This implies that variance $\tilde{\sigma}_{(u,v)}$ corresponding to link (u,v) is not uniform any longer but depends on the length of time interval (t_i, t_{i+1}) expanded by the two consecutive grid points over which the delay is reported. Therefore, finding an MLP is equivalent by solving the following optimization problem:

$$\tilde{R} = \mathrm{argmax}_R P(d(R) < T) = \mathrm{argmax}_R \Phi\left(\frac{T - \sum_{(u,v)\in R} m_{(u,v)}}{\sqrt{\sum_{(u,v)\in R} \sigma^2_{(u,v)}}}\right). \tag{6}$$

Unfortunately, one cannot derive an algorithm with polynomial complexity to minimize (6). Therefore, further elaboration of the method is needed. In order to tackle QoS routing, we need an extension of the Bellman Ford algorithm. Namely, we look for the optimal path R_{bf} from a source node $b \in V$ to a destination node $f \in V$ in a graph. Assign a triplet to each node $v \in V$ as

$$\mathcal{Q}_v = \{R_{bv}, m_{R_{bv}}, \sigma_{R_{bv}}\}, \tag{7}$$

where $m_{R_{bv}} := \sum_{(i,j)\in R_{bv}} m_{(i,j)}$, $\sigma_{R_{bv}} = \sqrt{\sum_{(i,j)\in R_{bv}} \sigma^2_{(i,j)}}$. The quality of a path is measured by the following quantity

$$P\left(d(R_{bv}) < T\right) := \Phi\left(\frac{T - \sum_{(u,v)\in R_{bv}} m_{(u,v)}}{\sqrt{\sum_{(u,v)\in R_{bv}} \sigma^2_{(u,v)}}}\right). \tag{8}$$

For short $N(m_{R_{bv}}, \sigma_{R_{bv}})$ denotes the Gaussian distribution of the overall delay along the path R_{bv} obtained by the convolution of the individual link delay distributions, given by (8).

"The General Normal Algorithm":
Initialize the algorithm at the initial step by setting $R_v(0) := \emptyset$, $m_{R_{bb}}(0) := 0$ and $\sigma_{R_{bb}}(0) := 0$ (the values of $m_{R_{bv}}(0)$ and $\sigma_{R_{bv}}(0)$ are indifferent) in $\mathcal{Q}_v$ for each $v \in V \backslash \{b\}$.
Then perform the following steps for all $v \in V \backslash \{b\}$ (which can be carried out parallel in every node).

1. Assume that we are in step k. Send the elements of $Q_v(k)$ to all the neighboring nodes w the set of which is denoted by $A_v = \{w : (v, w) \in E\}$. (As each node acts in the same manner, the elements of Q_w $(w \in A_v)$ are available at node v.)

2. Introduce a A'_v as $A'_v := A_v$ and two auxiliary variables: $p := 1$ and $z := v$.

3. Choose a $w \in A'_v$ and overwrite A'_v as $A'_v := A'_v \backslash \{w\}$. If $R_w \neq \emptyset$ or $w = b$ then

$$p_{acc} = 1 - \phi \left(\frac{T - m_{R_{bw}}(k) - m_{(w,v)}}{\sqrt{\sigma^2_{R_{bw}}(k) + \sigma^2_{(v,w)}}} \right).$$

   where $m_{(v,w)}$ and $\sigma_{(v,w)}$ are the parameters of the $\{v, w\}$ link. If $p_{acc} < p$ then $p := p_{acc}$ and $z := w$.

4. Repeat the step 4 until $A'_v = \emptyset$.

5. If $z \neq v$ and $p < 1 - \phi((T - m_{R_{bv}}(k))/\sigma_{R_{bv}}(k))$ then $m_{R_{bv}}(k+1) := m_{R_{bz}}(k) + m_{(v,z)}$, $\sigma_{R_{bv}}(k+1) = \sqrt{\sigma^2_{R_{bz}}(k) + \sigma^2_{(v,z)}}$ and $R_{bv}(k+1) := R_{bz}(k) \bigcup \{(z, v)\}$.

6. If $Q_v(k+1) = Q_v(k)$ for all $v \in V$ then STOP (the optimal path $\tilde{R}$ will be R_{bf}), else $k := k + 1$ and go to 2.

The proof of the convergence and the optimality of the algorithm given above boils down to extending the Bellman Ford method to abstract algebraic structures like monoids. Since the proof is quite involving we refer to [7, 5].

3.2 Finding an MLP by the Chernoff inequality

In this section, we develop QoS routing algorithm based on a large deviation approach which cast the search for MLP as a tail estimation problem. Namely, the objective function

$$\tilde{R}: \max_R P \left(\sum_{(u,v) \in R} \delta_{(u,v)} < T \right)$$

can be rewritten into

$$\tilde{R}: \min_R P \left(\sum_{(u,v) \in R} \delta_{(u,v)} > T \right) \tag{9}$$

which can give rise to the use of well-known statistical inequalities used for estimating the tail of the aggregated delay of a path.

Theorem 2:
Using the log moment generating function $\mu_{(u,v)}(s) = \log E\left(e^{s\delta_{(u,v)}}\right) = \log \int_{-\infty}^{\infty} f_{(u,v)}(s) e^{sy} dy$,

ARII can be reduced to SPR with link metric $\kappa_{(u,v)} := \mu_{(u,v)}(\hat{s})$, where $\hat{s} : inf_s \sum_{(u,v)\in\tilde{R}} \mu_{(u,v)}(s) - sT$.

Proof:
As was mentioned before, selecting a path, which maximizes the probability $P(\sum_{(u,v)\in R} d_{(u,v)} < T)$ is equivalent to selecting a path which minimizes $P(\sum_{(u,v)\in R} d_{(u,v)} > T)$. The latter probability, however, can be upperbounded by using the Chernoff bound

$$P(\sum_{(u,v)\in R} \delta_{(u,v)} > T) \leq e^{\mu_R(s)-sT}, \tag{10}$$

where $\mu_R(s)$ is the log-moment generating function of the aggregated delay, given as $\mu_R(s) = \sum_{(u,v)\in R} \mu_{(u,v)}(s)$. Therefore,

$$\tilde{R} : \min_R e^{\sum_{(u,v)\in R} \mu_{(u,v)}(s) - sT},$$

yields

$$\tilde{R} : \min_R \sum_{(u,v)\in R} \mu_{(u,v)}(s).$$

This last problem is an SPR with metric $\mu_{(u,v)}(s)$.
Q.E.D.

We term the method based on the Chernoff inequality with a given "s" as "Simple Chernoff Algorithm". One must note that we minimize a bound instead of the true objective function. Thus, the path found by this method can only be asymptotically optimal. More precisely, since

$$P(\sum_{(u,v)\in R} d_{(u,v)} < T) > 1 - e^{\mu_R(s)-sT}$$

one can state that the QoS requirement is met at least with probability $1-\epsilon$ where $\epsilon = e^{\mu_R(s)-sT}$. Thus this method can quantify the likelihood of satisfying the given QoS parameters, therefore may still prove to be useful from engineering point of view.
The other problem regarding the method is to find the proper value of s which yields the tightest bound. As was seen $\hat{s}$ depends on the path itself. Therefore, choosing an arbitrary s_1 and carrying out the corresponding BF algorithm, it may yield a false result (with another s_2 a totally different path can be found which might yield a better result. This gives rise to the following algorithm which term "exhaustive-s" algorithm:

Exhaustive-s algorithm:

1. Define a grid on the set of possible values of s denoted by $\mathcal{S} = \{s_i,\ s_i > 0,\ i = 1, ..., M\}$.
2. Pick an $s_i \in \mathcal{S}$
3. Perform SPR by the BF algorithm with link measures $\mu_{(u,v)}(s_i) := \log E\left(e^{s_i \delta_{(u,v)}}\right)$.
4. Based on the selected path $\tilde{R}_i$ determine

$$\hat{s}_i : \sum_{(u,v)\in\tilde{R}_i} \frac{d\mu_{(u,v)}(s)}{ds} = T \tag{11}$$

and calculate the bound

$$B_i := e^{\sum_{(u,v)\in\tilde{R}_i} \mu_{(u,v)}(\hat{s}_i) - \hat{s}_i T} \tag{12}$$

5. Find the path which belongs to minimal bound

$$\tilde{R}_j : \min_{R_i} e^{\sum_{(u,v)\in\tilde{R}_i} \mu_{(u,v)}(\hat{s}_i) - \hat{s}_i T} \tag{13}$$

It is obvious that the complexity of this algorithm is $O(M \mid V \mid^2)$ which can be overwhelming if M is large. Furthermore, since parameter s can take any positive value but grid $\mathcal{S}$ is finite, we may have missed the best path by simply ignoring some of the s parameters not being contained by grid $\mathcal{S}$.

The numerical complexity of the algorithm can be relaxed by assuming again an equidistant grid of delays $\mathcal{A}$ as we did initially. Let us suppose that over each interval (t_i, t_{i+1}) the delay follows the same discrete p.d.f. given as

$$P\left(\delta_{(u,v)} = a_{(u,v)} + m\Delta\right) = P_m, \tag{14}$$

where $m = -M, -M+1, ... -1, 0, 1, ..., M-1, M$, $a_{(u,v)} = \frac{t_i + t_{i+1}}{2}$ denoting the integer associated with link $(u,v) \in E$, which indicates that $\delta_{(u,v)} \in (t_i, t_{i+1})$ and $\Delta := \frac{t_{i+1} - t_i}{2M}$. Then $\mu_{(u,v)}$ is given as

$$\mu_{(u,v)}(s) = \log\left(\sum_{m=-M}^{M} P_m e^{s(m\Delta + a_{(u,v)})}\right), \tag{15}$$

which yields

$$\mu_{(u,v)}(s) = s a_{(u,v)} + \mu(s) \tag{16}$$

where

$$\mu(s) := \log\left(\sum_{m=-M}^{M} P_m e^{sm\Delta}\right) \tag{17}$$

is a link independent general logarithmic moment generation function. Based on this assumption, the following lemma can be stated:

Lemma 3:
For the QoS of a given path R the sharpest bound can be obtained as:

$$P\left(\sum_{(u,v)\in R} \delta_{(u,v)} > T\right) \le e^{\sum_{(u,v)\in R} \mu_{(u,v)}(\tilde{s}) - \tilde{s}T}, \tag{18}$$

where

$$\tilde{s} = \frac{\mu'^{-1}\left(T - \sum_{(u,v)\in R} a_{(u,v)}\right)}{\mid R \mid}. \tag{19}$$

In the expression above $\mu'^{-1}(x)$ is the inverse of the derivative of the link independent logarithmic moment generating function $\mu(s)$ and $\mid R \mid$ indicates the number of hops in path R.

Proof:
To obtain the sharpest bound we have to optimize s as follows:

$$\tilde{s} : \inf_s \sum_{(u,v)\in R} \mu_{(u,v)}(s) - sT.$$

This can be achieved by differentiation (see [6, 8]), yielding

$$\tilde{s}: \inf_s \sum_{(u,v)\in R} \frac{d\mu_{(u,v)}(s)}{ds} = T.$$

Taking into account that $\mu_{(u,v)}(s) = sa_{(u,v)} + \mu(s)$ and performing the differentiation, one can obtain

$$\sum_{(u,v)\in R} \mu'(\tilde{s}) = T - \sum_{(u,v)\in R} a_{(u,v)},$$

which indeed yields

$$\tilde{s} = \frac{\mu'^{-1}\left(T - \sum_{(u,v)\in R} a_{(u,v)}\right)}{\mid R \mid}.$$

Q.E.D.

Based on the lemma above one can easily calculate the $\tilde{s}$ which belong to path R. To illustrate this dependence, we will use the notation $\tilde{s}(R)$ in the forthcoming discussion.
Due to the easy optimization of s according to Lemma 3, another method can be proposed to find an MLP, which we term as "Recursive Path Finder - s Finder Algorithm". The name originates from the fact that with a given s we find an optimal path $R(s)$ then for this path we determine $\tilde{s}(R)$ and search for a new path ...etc.

<u>The Recursive Path Finder - s Finder Algorithm:</u>
Pick a positive s and carry out the following steps recursively:

1. Associate measure $\mu_{(u,v)}(s)$ to each link $(u,v) \in E$.
2. Perform the BF algorithm to find the optimal path $\tilde{R}(s)$ for parameter s.
3. For the obtained $\tilde{R}$ determine $\tilde{s}$ by expression $\tilde{s} = \frac{\mu'^{-1}\left(T - \sum_{(u,v)\in R} a_{(u,v)}\right)}{|R|}$.
4. Go back to Step 1 and perform BF algorithm again by using $\tilde{s}$.

From the algorithm above it is clear that in each step either the quality of the path is improved (for a given s we find an optimal $\tilde{R}$) or the quality of the bound is improved (for a given path we find an optimal $\tilde{s}$). Therefore, by carrying out the algorithm recursively, we always improve the solution. The algorithm will settle in a fix point if the following equation holds :

$$\tilde{R}(\tilde{s}) = \tilde{R}(s), \tag{20}$$

meaning that for a given s the optimal path $\tilde{R}$ remains the same as the path provided by the BF algorithm which is run again with the optimized $\tilde{s}(\tilde{R})$. We must remark that this algorithm was only tested by simulation, the theoretical proof of the existence and uniqueness of such steady state remains to be the task of future work.

The method given above relies on the fact that the optimal $\tilde{s}$ can be calculated for a given path R relatively simply. This can be done easily if the log moment generating function of link delays can be given as $\mu_{(u,v)}(s) = sa_{(u,v)} + \mu(s)$ where $\mu(s)$ is the link independent part (see (17)), which assumes equidistant grid $\mathcal{A}$. In the case of non-equidistant grid a further generalization of the algorithm is needed, which is summarized in the next algorithm:

<u>The "General MLP Selection Algorithm by Chernoff Inequality":</u>

For each node $(u,v) \in E$ the logarithmic moment generating function of link delay $\mu_{(u,v)}(s)$ is given (which is a general function and not necessarily be given in the form of (16)). Let us assume that we would like to find a path from node b to node f. Furthermore, we are at stage k of having a k-hop subpath $R(k)$ starting from node b and ending with m and the task is find out which node to connect next. Let the set of neighboring nodes to node n be denoted by V_m. Then carry out the following steps, recursively.

1. Determine $s_n : \min_s \sum_{(u,v)\in R(k)} \mu_{(u,v)}(s) + \mu_{(m,n)}(s) - sT$ for each $n \in V_m$.
2. Select node i for which $i = \operatorname{argmin}_n \sum_{(u,v)\in R(k)} \mu_{(u,v)}(s_n) + \mu_{(m,n)}(s_n) - s_n T$.
3. Extend the path as $R(k+1) = R(k) \bigcup \{(m,i)\}$ and go back to Step 1.

Theorem 3:
An MLP can be selected in polynomial time by using the "General MLP Selection Algorithm by Chernoff Inequality".

Proof :
The proof here is based on again extending the BF algorithm to monoids [7, 5]. The key notion in the proof is to ascertain the underlying optimality principle, given as follows: $\kappa(R(k)) \leq \kappa(R(k+1))$. Or more precisely $\kappa(R(k)) \leq \kappa(R(k)\bigcup\{(m,i)\})$, where $\kappa(R) = \sum_{(u,v)\in R} \mu_{(u,v)}(\tilde{s}) - \tilde{s}T$ and $\tilde{s} : \min_s \sum_{(u,v)\in R} \mu_{(u,v)}(s) - sT$. For this we have to prove that $\min_s \sum_{(u,v)\in R} \mu_{(u,v)}(\tilde{s}) - \tilde{s}T \leq \min_s \sum_{(u,v)\in R} \mu_{(u,v)}(\tilde{s}) + \mu_{(m,n)}(s) - \tilde{s}T \quad \forall n \in V_m$. This can be pointed out rather straightforwardly by using the fact that the logarithmic moment generating function $\mu(s)$ of a positive random variable (link delay) is always positive. Therefore adding an additional term $\mu_{(m,n)}(s) \geq 0$ to the expression $\sum_{(u,v)\in R} \mu_{(u,v)}(\tilde{s}) - \tilde{s}T$ it will hold that

$$\sum_{(u,v)\in R} \mu_{(u,v)}(\tilde{s}) - \tilde{s}T \leq \sum_{(u,v)\in R} \mu_{(u,v)}(\tilde{s}) + \mu_{(m,n)}(s) - \tilde{s}T \;\; \forall s > 0,$$

which implies that

$$\min_s \sum_{(u,v)\in R} \mu_{(u,v)}(\tilde{s}) - \tilde{s}T \leq \min_s \sum_{(u,v)\in R} \mu_{(u,v)}(\tilde{s}) + \mu_{(m,n)}(s) - \tilde{s}T.$$

The rest of the proof is done similarly to the case of "General Normal Algorithm" and the interested reader can find it in [5].

4 Performance analysis and simulation results

In this section the performances of the newly developed QoS routing algorithms are analyzed by extensive simulations. In order to compare the algorithms we introduced a performance measure for the paths starting from node b and ending at node f denoted by $\eta(b,f)$. This is defined as the ratio of the probability of the path (found by a given algorithm) fulfilling the end-to-end QoS criterion and the probability of the path (found by exhaustive search) fulfilling the end-to-end QoS criterion, given as follows :

$$\eta(b,f) := \frac{P\left(\sum_{(u,v)\in R_{\text{found by a given algorithm}}} \delta_{(u,v)} < T\right)}{P\left(\sum_{(u,v)\in R_{\text{found by exhaustive search}}} \delta_{(u,v)} < T\right)}. \tag{21}$$

However, one can calculate $\eta(b,f)$ for a whole graph (i.e., for each possible starting node b and for each possible ending node $f \in E$), yielding the following performance function $\eta(T)$:

$$\eta(T) := \frac{\sum_{b\in V}\sum_{f\in V} P\left(\sum_{(u,v)\in R(b,f)_{\text{found by a given algorithm}}} < T\right)}{\sum_{b\in V}\sum_{f\in V} P\left(\sum_{(u,v)\in R(b,f)_{\text{found by exhaustive search}}} < T\right)}. \tag{22}$$

In the case of a given graph this measure only depends on the value of the actual end-to-end QoS requirement. The closer this function approximates the value 1, the better the performance of the corresponding routing algorithm is. In order to quantify this approximation error we introduce the following quantity:

$$\chi := \frac{1}{D} \int_{T=0}^{D} (1 - \eta(T))^2 \, dT, \tag{23}$$

where $D : P\left(\sum_{(u,v)\in R} \delta_{(u,v)} < D\right) = 1$. It is easy to see that the better the method is, the closer the value of χ falls to 0.

Figure 1, depicts the test network (which has a similar topology to a Local Exchange Network network), we used to simulate the performance of the different routing algorithms.
Figure 2 depicts $\eta(T)$ in the case of assuming normally distributed link delays and when routing

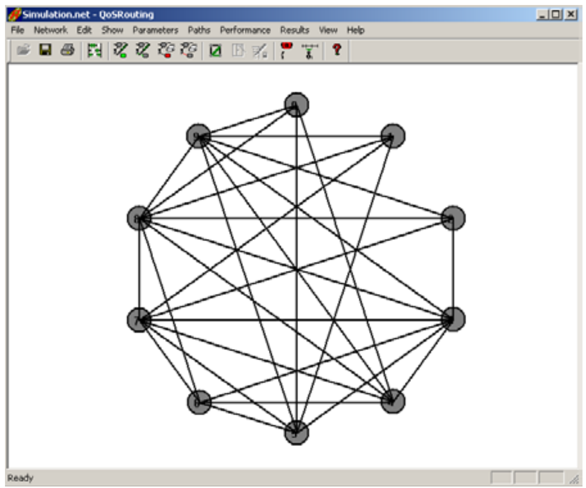

Figure 1.: *The topology of the test network*

is carried out by the Simple Normal and General Normal algorithms, respectively.
On can see the performance of each method is very high since the efficiency never goes bellow

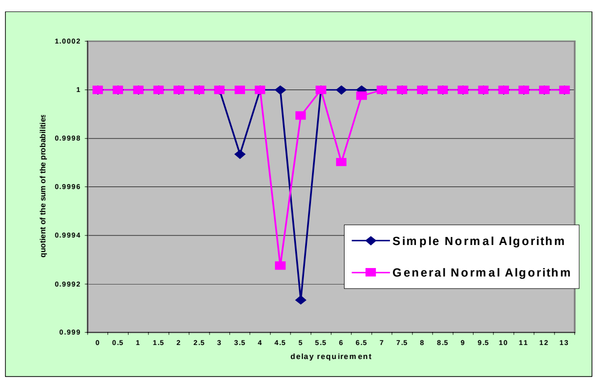

Figure 2.: *The performance as a function of end-to-end delay requirement in the case of Simple Normal and General Normal Algorithms*

0.999. On Figure 3, one can see the averaged quadratic errors achieved by the "Simple Normal" and "General Normal" algorithms.

As was expected the "General Normal Algorithm yields better performance than the Simple Normal Algorithm which is due to the fact that the latter one cannot take into consideration

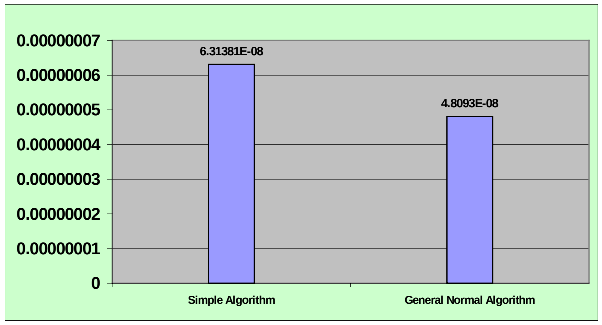

Figure 3.: *The overall Quadratic error of the Simple Normal and General Normal Algorithms*

variable size time intervals.
Figure 4 shows $\eta(T)$ in the case of assuming Bernoulli distributed link delays and when routing is carried out by the Simple Chernoff algorithm with variable parameters $s = 1.5, 2, 5, 10$.
Based on the corresponding χ parameters, Figure 5 demonstrates that $s = 1.5$ yields a consid-

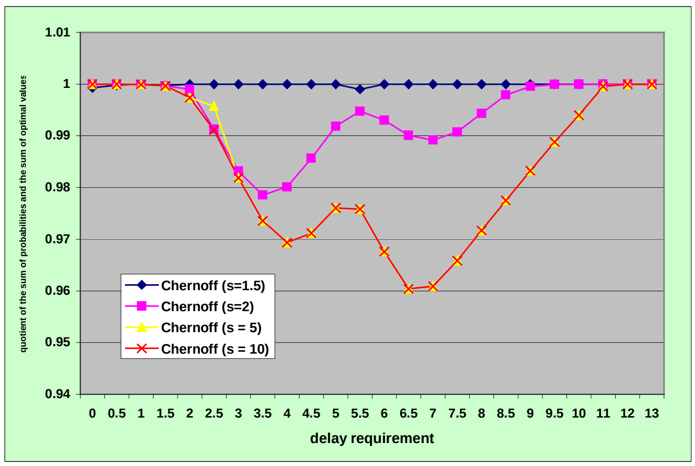

Figure 4.: *The performance as a function of end-to-end delay requirement in the case of the Simple Chernoff Algorithm with given "s" parameters*

erably better path than $s = 5$ or $s = 10$.
This reveals the sensitivity of routing performance with respect to s. Figure 6 shows $\eta(T)$ in the case of "Exhaustive-s", "Recursive Path Finder - s Finder" and "General Chernoff" algorithms, respectively, while Figure 7 exhibits the corresponding quadratic errors.

As can be seen "The General Chernoff Algorithm" yields the best performance amontge the CHrnoff type methods due to the fact that it can be used in the case of non-equidistant delay grid, as well.

In order to obtain more general numerical results, we run the routing algorithms not only on one particular graph, but on a set of graphs (denoted by $\mathcal{G}$) with 10 nodes, but the links and the corresponding delays were generated randomly . Consequently, we introduced the following "ensemble" performance measures $\eta_e(T)$ and χ_e, given as follows:

$$\eta_e(T) := \frac{\sum_{graph\in\mathcal{G}} \sum_{b\in V} \sum_{f\in V} P\left(\sum_{(u,v)\in R(b,f)\text{found by a given algorithm}} \delta_{(u,v)} < T\right)}{\sum_{graph\in\mathcal{G}} \sum_{b\in V} \sum_{f\in V} P\left(\sum_{(u,v)\in R(b,f)\text{found by exhaustive search}} \delta_{(u,v)} < T\right)}, \quad (24)$$

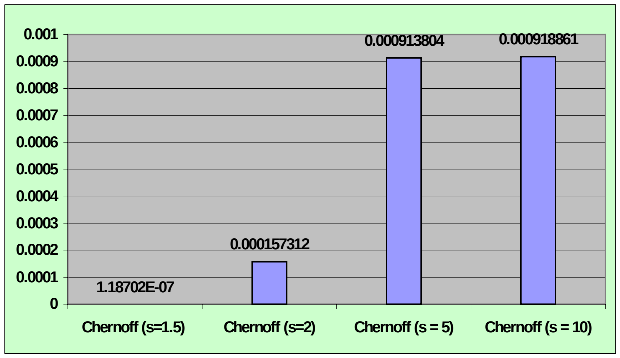

Figure 5.: *The overall quadratic error of the Simple Chernoff Algortihm with given "s" parameters*

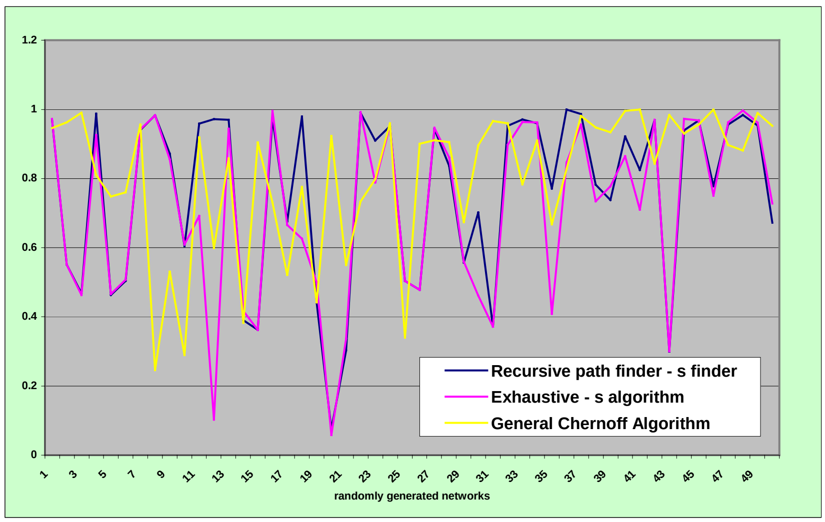

Figure 6.: *The performance over a randomly generated ensemble as a function of end-to-end delay requirement in the case of Recursive path finder - s finder, Exhaustive - s and General Chernoff Algorithms*

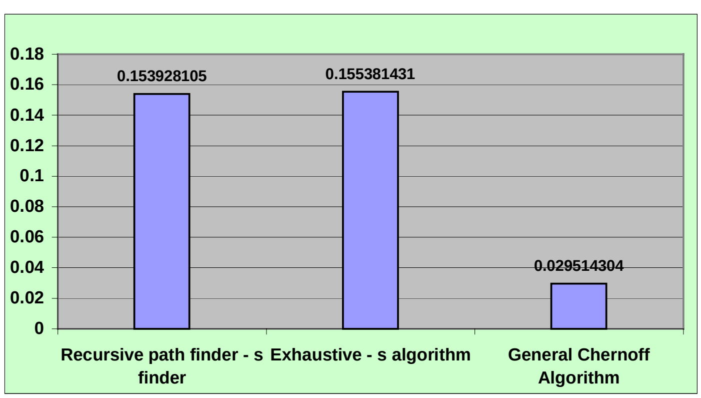

Figure 7.: *The overall quadratic error over a randomly generated graph ensemble in the case of Recursive path finder - s finder, Exhaustive - s and General Chernoff Algorithms*

$$\chi_e := 1 - \frac{1}{D} \int_{T=0}^{D} (1 - \eta_e(T))^2 \, dT. \tag{25}$$

Figure 8 shows $\eta_e(T)$ achieved by the "Simple Normal" and "General Normal" algorithms, respectively, while Figure 9 exhibits the corresponding quadratic errors.

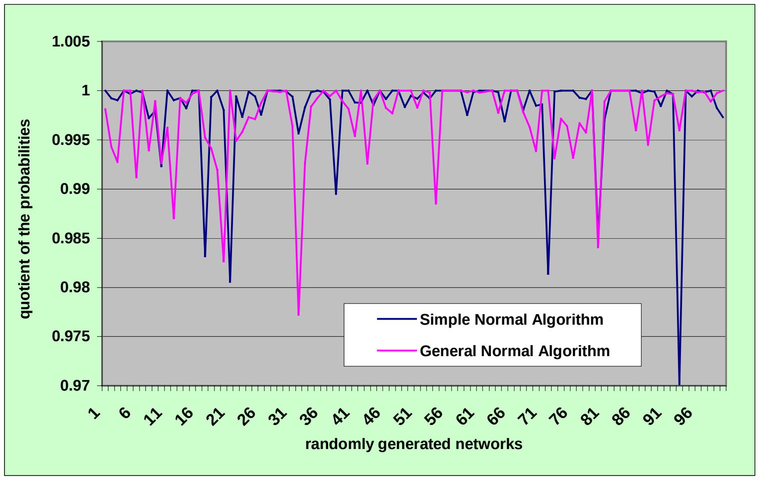

Figure 8.: *The performance over a randomly generated ensemble as a function of end-to-end delay requirement in the case of Simple Normal, General Normal and Simple Chernoff Algorithms*

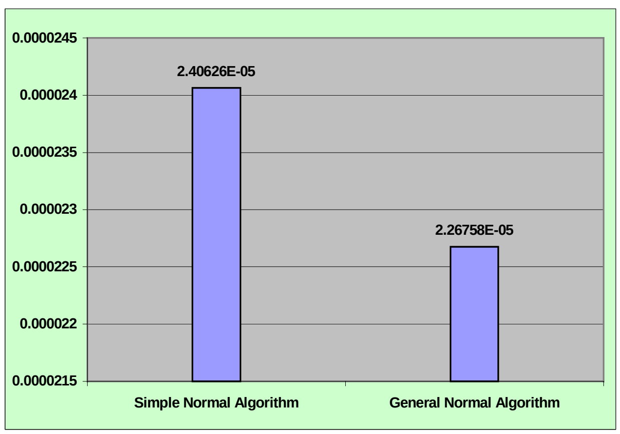

Figure 9.: *The overall quadratic error over a randomly generated graph ensemble in the case of Simple Normal, General Normal and Simple Chernoff Algorithms*

One can see that again the "General Normal Algorithm" proves to be the best. Figure 10 and 11 summarizes the performance curves and the quadratic errors of the Chernoff type methods over the randomly generated graph ensemble.

Again the "General Chernoff Algorithm" proves to the best amont the Chernoff type methods over the randomly generated graph ensemble.

Based on the simulation results we can rank the developed algorithms with respect to their performances as follows:

1. "General Normal Algorithm"
2. "Simple Normal Algorithm"

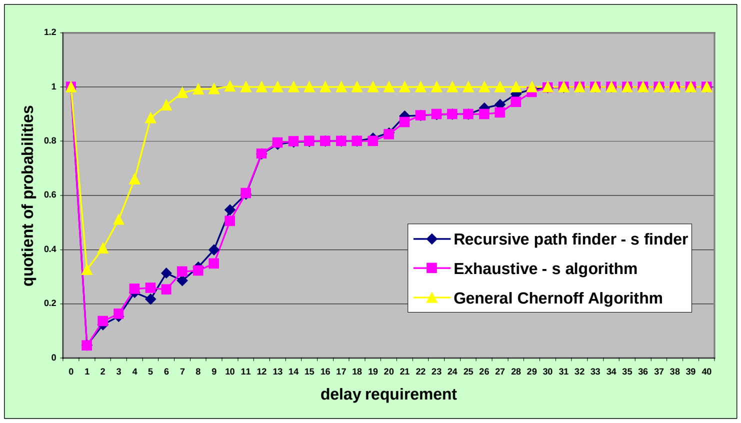

Figure 10.: *The performance as a function of end-to-end delay requirement in the case of Recursive path finder - s finder, Exhaustive - s and General Chernoff Algorithms*

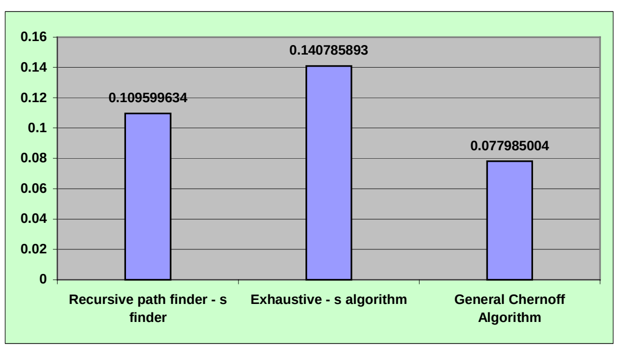

Figure 11.: *The overall quadratic error of the Recursive path finder - s finder, Exhaustive - s and General Chernoff Algorithms*

3. "General Chernoff Algorithm"

4. "Recursive Path Finder - s Finder Algorithm"

5. "Exhaustive-s Algorithm"

Therefore, either the "General Normal Algorithm" (the best performance but a bit higher numerical complexity) or the "Simple Normal Algorithm" (a slight decrease in performance but smaller numerical complexity) is recommended for practical applications.

5 Conclusions

New algorithms were proposed in the paper to carry out QoS routing with incomplete information. The proposed algorithms are capable of carrying out routing in polynomial time. Based on the theoretical and numerical analysis the best method is the General Normal algorithm, however, methods based on the Chernoff inequality also provide good performance.

6 Acknowledgement

The research reported here was carried out in the framework of High Speed Networks Laboratory.

References

[1] Cherukuri, R., Dykeman, D.: "PNNI draft specification", *ATM Forum* 94-0471, November, 1995

[2] Lorenz, D., Orda, A.: "QoS routing in networks with uncertain information", *IEEE/ACM Trans. Networking*, vol. 6., December, 1998.

[3] Lee, W.: "Spanning tree methods for link state aggregation in large communication networks", *Proc. INFOCOM*, Boston, MA, April, 1995

[4] Guerin, R., Orda, A.: "QoS routing in networks with inaccurate information: theory and algorithms", ", IEEE/ACM Trans. Networking, vol. 7., June, 1999.

[5] Fancsali, A., Levendovszky, J.: "The extension of Shortest Path Algorithms to abstract algebraic structures" *Tehcnical Report*, Departament of Telecommunications, Budapest University of Technology and Economics, January, 2001.

[6] Levendovszky, J, E.C. van der Meulen,: "Tail Distribution Estimation for Call Admission Contorl in ATM Networks", *Proccedings of IFIP, Third Workshop on Performance Modelling and Evaluation of ATM Networks*, Ilkley, West Yorkshire, UK, 2-6th July 1995.

[7] G. Birkhoff and T. C. Bartee, *Modern Applied Algebra*, New York, NY: McGraw-Hill, 1970.

[8] Hui, J.: *"Resource Allocation for Broadband Networks"*, IEEE Journal on Selected Area of Communications, Vol.6, No.9., pp. 1598-1608, December 1988.

[9] Crawley, E., Nair, R., Rajagopalan, B., Sandick., H.: "A Framework for QoS-based Routing in the Internet" *RFC 2386*, Argon Networks, Arrowpoint, NEC USA, Bay Networks, August 1998.

[10] Guerin, R., Kamat, S., Orda, A., Przygienda T., Williams, D.: "QoS Routing Mechanisms and OSPF Extensions" *RFC 2676*, IBM, Technion, Lucent, December 1998.

[11] Apostolopoulos, Guerin, Kamat, Tripathi: "Quality of Service Based Routing: A Performance Perspective", *Proceedings of SIGCOM*, pp. 17-28, Vancouver, Ontario, Canada, September 1998.

[12] Wang, Z., Crowcroft, J.: "Quality-of-Service Routing for Supporting Multimedia Applications", *IEEE Journal on Selected Areas in Communications*, vol. 14, pp. 1228-1234, September 1996

[13] PNNI Specification Working Group: "Private Network-Network Interface Specification Version 1.0", *ATM Forum*, March 1996, *available at ftp://ftp.atmforum.com/pub/approved-specs/af-pnni-0055.000*

[14] Jaffe, J.:"Algorithms for Finding Paths with Multiple Constraints" Networks, Vol 14, pp 95-116, 1984

[15] Shaikh, A., Rexford, J., Shin, K.: "Dynamics of Quality-of-Service Routing with Inaccurate Link-State Information," *Technical Report CSE-TR-350-97*, Computer Science and Engineering Division, Dept. of Electrical Engineering and Computer Science, University of Michigan, November 1997

[16] Shaikh, A., Rexford, J., Shin, K.: "Efficient Precomputation of quality-of-service routes" *Proceedings of Workshop on Network and Operating System Support for Digital Audio and Video*, July 1998.