Quality-of-Service Routing for Traffic with Performance Guarantees

Qingming Ma and Peter Steenkiste
Computer Science Department, Carnegie Mellon University
Pittsburgh, PA 15213, USA, {qma, prs}@cs.cmu.edu

Abstract

Quality-of-Service (QoS) routing tries to select a path that satisfies a set of QoS constraints, while also achieving overall network resource efficiency. We present initial results on QoS path selection for traffic requiring bandwidth and delay guarantees. For traffic with bandwidth guarantees, we found that several routing algorithms that favor paths with fewer hops perform well. For traffic with delay guarantees, we show that for a broad class of WFQ-like scheduling algorithms, the problem of finding a path satisfying bandwidth, delay, delay-jitter, and/or buffer space constraints while at the same time deriving the bandwidth that has to be reserved to meet these constraints, is solvable by a modified version of the Bellman-Ford shortest-path algorithm in polynomial time.

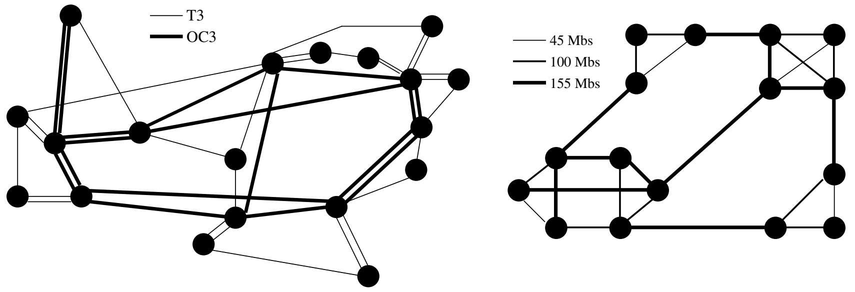
Keywords

Routing, quality of service, integrated services networks

1 INTRODUCTION

Future integrated-service packet-switching networks (ISPN) will support a variety of service classes to meet diverse quality-of-service (QoS) requirements of existing and emerging data and multimedia applications. Among the service models being proposed, two classes are of particular interest: traffic with bandwidth guarantees and traffic with stringent end-to-end delay bounds. The former includes the controlled-load (IETF) and available-bit-rate with minimal cell rate (ATM Forum) services. The latter includes the guaranteed service (IETF) and the constant- and variable-bit-rate services (ATM Forum).

QoS routing is the first step toward achieving end-to-end QoS guarantees. It identifies paths that meet QoS constraints, and selects one that leads to high overall resource efficiency. In this paper, we present initial results on routing traffic with bandwidth guarantees (Section 2) and latency guarantees (Section 3). We summarize in Section 5.



(a) MCI topology (b) Clustered Topology

Figure 1 Topologies

2 BANDWIDTH GUARANTEES

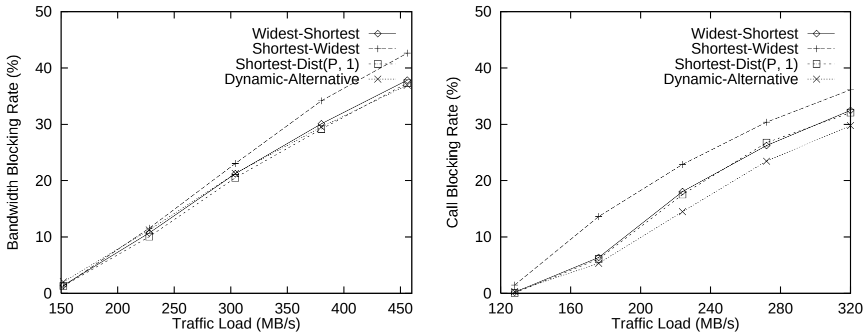
For services with bandwidth guarantees, QoS routing tries to identify a *feasible path*, i.e. a path on which all links have an unserved bandwidth that is higher than the requested bandwidth. Several path selection algorithms have been proposed (see Section 4), but a systematic evaluation of algorithms is missing. In this section we present initial results of a simulation study comparing the following four algorithms:

- **Widest-shortest path**: a feasible path with the minimum hop count. If there are several such paths, the one with the maximum bandwidth is selected. If several such paths exist, one is randomly selected.
- **Shortest-widest path**: a feasible path with the maximum bandwidth. If there are several such paths, the one with the minimum hop count is selected. If several such paths exist, one is randomly selected.
- **Dynamic-alternative path**: a widest minimal-hop path. If no feasible minimal hop path exists, find the widest path that is one hop longer. If several such paths exist, one is randomly selected. Reject the request otherwise.
- **Shortest-**dist$(P, 1)$: a path with the shortest distance

  $$\textbf{Shortest}\text{-dist}(P,1) = \sum_{i=1}^{k} \frac{1}{R_i}$$

  where $R_1, \cdots, R_k$ are the bandwidths available on the links on path P. This algorithm has been shown to be effective when selecting routes for high-bandwidth connections (Ma et.al, 1996).

Our study uses two topologies (Figure 1). The traffic load is a combination of audio and video sessions. We assume that the requested bandwidth is uniformly distributed between 16 $\sim$ 64 kilobits/second for an audio session,



(a) MCI: 60%/40% Audio/Video (b) Clustered: All Video

Figure 2 Blocking rate as a function of an evenly distributed network load

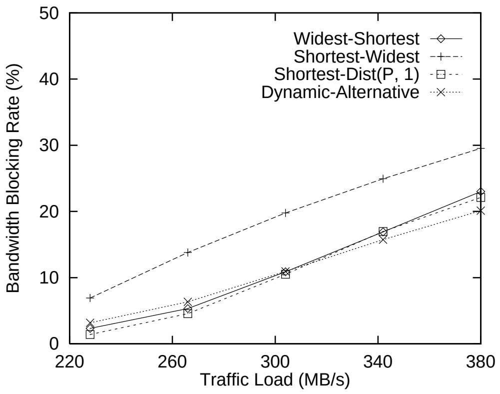


Figure 3 Call blocking rate as a function of network load: MCI topology, 100% Video traffic, Unevenly distributed load

and between $1 \sim 5$ megabits/second for a video session. Sessions have a Poisson arrival rate. Based on Bolotin (1994), we use a lognormal long-tail call holding time distribution. All four algorithms use the dynamic link state information (residual bandwidth) with a refresh rate of 30 seconds. A common performance metric for traffic with bandwidth guarantees metric is the Call Blocking Rate, the percentage of session requests that is rejected. This metric is however misleading if sessions can request different amounts of bandwidth. Thus, we introduce a new metric, the *Bandwidth Blocking Rate*, which takes the session bandwidth into account:

$$\text{bandwidth blocking rate} = \frac{\sum_{i \in \mathcal{B}} \text{bandwidth}(i)}{\sum_{i \in \mathcal{S}} \text{bandwidth}(i)}$$

where $\mathcal{B}$ is the set of blocked sessions and $\mathcal{S}$ the set of requested sessions.

Figure 2 shows the bandwidth blocking rate as a function of network load for the case that the traffic is evenly distributed. Figure 3 shows the result for an uneven load, where most of the traffic is between the East and West Coast. In all cases, we observe that most algorithms achieve similar performance. The exception is the shortest-widest algorithm, which tends to pick longer path, and is therefore more resource intensive.

This result is different from that obtained for best effort traffic by Ma et.al. (1996), where the shortest distance path gave overall the best performance. The difference is that with best effort traffic, all paths are feasible, even paths that are heavily congested relative to other parts of the network. The shortest distance algorithm was able to route around congested links more effectively than, for example, the widest-shortest path. However, when bandwidth is reserved, heavily congested links will no longer be feasible, so any algorithm will route around them, and algorithms that favor short paths will give similar performance and have efficient resource utilization.

3 DELAY GUARANTEES

QoS constraints for traffic requiring delay guarantees include end-to-end delay, delay-jitter, and buffer space bounds. Most existing QoS routing studies assume these QoS constraints are independent, and a well-known result is that finding a path with (independent) bandwidth, delay, and delay-jitter constraints is NP-complete (Wang and Crowcroft, 1996; Garey and Johnson, 1979). In practice these bounds are functions of the reserved bandwidth, the selected path, the traffic characteristics, and the switch scheduling algorithm, and they are not independent. We show that the problem of finding a path satisfying bandwidth, delay, delay-jitter, and buffer space constraints can be simplified by taking these relationships into consideration,

Recent studies in defining scheduling disciplines to support end-to-end delay guarantees have identified a class of rate-proportional scheduling algorithms (Zhang, 1995 and its references), including Virtual Clock, Weighted Fair Queueing, Worst-case Weighted Fair Queueing, and Self Clocked Fair Queueing. These WFQ-like scheduling algorithms isolate each guaranteed session from other sessions to ensure a guaranteed share of link resources. The queueing delay of the session is thus determined by the bandwidth being reserved and the burstiness of the traffic source. In this section, we show that in networks that use these WFQ-like scheduling algorithms, finding a path that satisfies delay, delay-jitter, and buffer space constraints is solvable in polynomial time if we take into consideration the relationship between the bandwidth, the delay, and the delay-jitter for this class of scheduling algorithms. The bandwidth to be reserved does not have to be known a priori.

3.1 Delay, Delay-Jitter, and Buffer Space

Assume that the traffic source is constrained by a token bucket $\langle \sigma, b \rangle$, where σ is the average token rate and b is token bucket size. For a given path $\mathbf{p}$ with n hops and the link capacity C_i, the provable end-to-end delay bound is given by (Zhang, 1995)

$$D(\mathbf{p}, r, b) = \frac{b}{r} + \frac{n \cdot L_{\max}}{r} + \sum_{i=1}^{n} \frac{L_{\max}}{C_i} + \sum_{i=1}^{n} \mathrm{prop}_i \tag{1}$$

where r ($r \geq \sigma$) is the bandwidth to be reserved, $L_{\max}$ is the maximal packet size in the network, and prop_i is the propagation delay. The end-to-end delay-jitter bound and buffer space requirement at the h-th hop are given by

$$J(\mathbf{p}, r, b) = \frac{b}{r} + \frac{n \cdot L_{\max}}{r}. \tag{2}$$

$$B(\mathbf{p}, b, j) = b + h \cdot L_{\max}. \tag{3}$$

3.2 Path Selection

A path is *feasible* for traffic with delay guarantees, if it meets the delay, delay-jitter, and buffer space requirements given in the Equations 1, 2, and 3, respectively. There are two cases to consider. First, the bandwidth r to be reserved is known *a priori*. Second, r is not known and has to be calculated by the routing algorithm. The main results of this section are summarized in the following theorem:

Theorem 1 *The QoS routing problem of finding a path with delay, delay-jitter, and/or buffer space constraints is solvable in polynomial time. If the bandwidth to be reserved is known* a priori, *a slightly modified version of the Bellman-Ford algorithm can solve it in $S = O(m \cdot L)$, where m is the number of nodes and L the number of links in the network. If the bandwidth to be reserved is unknown, an algorithm that iterates the modified version of the Bellman-Ford algorithm can solve it in $E \cdot S$, where E is the number of all possible residual bandwidth of links in the network. Moreover, if several paths are feasible, we can select one with any one of the following properties: minimum delay, minimum delay-jitter, or minimum hop count.*

Note that the maximal buffer space requirement of a path is determined by the path hop count. Thus, selecting a path with the minimum hop count reduces the maximal buffer space consumption. The following lemma will be frequently referenced in our discussion of path selection algorithms.

Lemma 1 *Given a path-length function $l(P) = \sum_{i \in P} l(i)$ with $l(i) > 0$ for all links i, a bound $\mathbf{d}$, and a hop bound N. Finding a path P from a source s to a destination d with $l(P) \leq \mathbf{d}$ and no more than N hops can be solved by the Bellman-Ford Algorithm in $O(N \cdot E)$, where E is the number links in G. Moreover, we can identify from all feasible paths a path with the minimum hop count, or with the minimum length.*

Proof. The Bellman-Ford algorithm (Bertsekas and Gallager, 1987) finds a shortest path step by step with increasing hop count: At the i-th step, a shortest path with at most i hops is found. The total number of steps is restricted to $\min\{m, N\}$, where m is the number of nodes in the network. The first feasible path found is the one with the minimum hop count. To find a path with the minimum length, we remember the paths found during each step and select the one with the minimum path length. □

(a) Delay Bound

The problem of finding a path satisfying a given end-to-end delay bound is formulated as follows.

Path Finding Problem (D-r): *Given a delay bound $\mathbf{d}$, a leaky bucket $\langle \sigma, b \rangle$, and a bandwidth r ($r \geq \sigma$) to reserve, find a path $\mathbf{p}$ with $r \leq R_j$ ($\forall j \in \mathbf{p}$), such that $D(\mathbf{p}, r, b) \leq \mathbf{d}$, where R_j is the residual bandwidth of link j.*

Path Finding Problem (D): *Given a delay bound $\mathbf{d}$ and a leaky bucket $\langle \sigma, b \rangle$, find a path $\mathbf{p}$, such that $D(\mathbf{p}, r, b) \leq \mathbf{d}$ for some r with $\sigma \leq r \leq R_j$ ($\forall j \in \mathbf{p}$), where R_j is the link residual bandwidth.*

Proposition 1 *QoS routing problem* **(D-r)** *is solvable by both the Dijkstra and Bellman-Ford shortest-path algorithms. Moreover, if there are feasible paths, we can select one with the minimum delay. If the Bellman-Ford algorithm is used, we can select a feasible path with either the minimum delay or the minimum hop count.*

Proof. Find a shortest path P using either Dijkstra or Bellman-Ford algorithm (considering only those links with $R_i \geq \sigma$) using the length function:

$$l(i) = \frac{L_{\max}}{r} + \frac{L_{\max}}{C_i} + \mathrm{prop}_i \quad \& \quad l(P) = \frac{b}{r} + \sum_{j \in P} l(j) \tag{4}$$

If $l(P) > \mathbf{d}$, there is no path that meets the delay bound $\mathbf{d}$. Otherwise, the P is a path with the minimum delay. If the Bellman-Ford algorithm is used, the first feasible path found is the one with the minimum hop count. □

Proposition 2 *QoS routing problem* **(D)** *is solvable by iterating any single-*

pair shortest-path algorithm over all possible residual link-bandwidth in $E \cdot S$ time, where E is the number of possible residual link-bandwidths and S is the time for the shortest-path algorithm. If several paths are feasible, we can select one with the minimum delay. If the Bellman-Ford shortest-path algorithm is used in each iteration, we can select a feasible path with either the minimum delay or the minimum hop count.

Proof. The difference with Proposition 1 is that the bandwidth r to be reserved is unknown. For a given path P, the delay can be reduced by increasing r. The maximal reservable bandwidth on path P is $\min\{R_j \mid j \in P\}$.

One would expect to use a shortest path algorithm using a length function as in Equation 4, setting r to the maximal reservable bandwidth on the partial path. The problem is that the maximal reservable bandwidth changes during the search. An earlier short path may turn into a long path when a link with small residual bandwidth is added to the path. To overcome this, we iterate the shortest-path algorithm over possible choices of link residual bandwidth. At each iteration, a fixed r is used in the length Equation 4, and only those links whose residual bandwidth are equal to or higher than r are considered. That is, for every $r = R_k$ of some link k in the network, we define a length function l_r as follows:

$$l_r(i) = \frac{L_{\max}}{r} + \frac{L_{\max}}{C_i} + \mathrm{prop}_i \quad \& \quad l_r(P) = \frac{b}{r} + \sum_{j \in P} l_r(j) \tag{5}$$

Using any shortest-path algorithm, we find a shortest path P_r from the source s to the destination d, such that there is no link j in P_r whose residual bandwidth R_j is less than r. We store r, P_r and $l_r(P_r)$ in a vector with E entries. After iterating r over all possible R_k, we search the whole vector to find a path $P_{\min}$ whose length $l_r(P_{\min})$ is minimal. We claim that the path $P_{\min}$ is the shortest delay path if $r = \min\{R_j \mid j \in P_{\min}\}$ is reserved. If it is not, there must be a path P' with a shorter delay than $P_{\min}$. Let r' be $\min\{R'_j \mid j \in P\}$, the maximal reservable bandwidth of the path P'. This r' must be the residual bandwidth of some links along the path P', and the residual bandwidth of all other links on P' must be no less than r'. This is impossible since $P_{r'}$ stored in the vector is a shorest delay path with $r' \leq R_j$.

If the Bellman-Ford shortest-path algorithm is used to iterate over every r, we can select a feasible path with minimal hop count whose residual bandwidth is at least r. A feasible path with the minimum hop count can be selected from the result paths of all iterations. □

(b) Delay and Delay-Jitter Bounds

The problem of finding a path satisfying both end-to-end delay and delay-jitter bounds can be formulated as:

Path Finding Problem (D-J-r): Given a delay bound $\mathbf{d}$, a delay-jitter bound $\mathbf{j}$, a leaky bucket $\langle \sigma, b \rangle$, and a bandwidth r ($r \geq \sigma$) to reserve, find a path $\mathbf{p}$ with $r \leq R_j (\forall j \in \mathbf{p})$, $D(\mathbf{p}, r, b) \leq \mathbf{d}$, and $J(\mathbf{p}, r, b) \leq \mathbf{j}$, where R_j is the residual bandwidth of link j.

Path Finding Problem (D-J): Given a delay bound $\mathbf{d}$, a delay-jitter bound $\mathbf{j}$, and a leaky bucket $\langle \sigma, b \rangle$, find a path $\mathbf{p}$, such that $D(\mathbf{p}, r, b) \leq \mathbf{d}$ and $J(\mathbf{p}, r, b) \leq \mathbf{j}$ for some r with $\sigma \leq r \leq R_j$ $(\forall j \in \mathbf{p})$, where R_j is the residual bandwidth of link j.

Proposition 3 *QoS routing problem* **(D-J-r)** *is solvable by the Bellman-Ford algorithm in time $m \cdot L$, where m is the number of nodes and L the number of links in the network. Moreover, if several paths are feasible, we can select one with any one of the following properties: minimum delay, minimum delay-jitter, or minimum hop count.*

Proof. We learn from Equation 2 that the hop count (n) is the only parameter that determines whether a delay-jitter bound $\mathbf{j}$ can be met:

$$\frac{b}{r} + \frac{n \cdot L_{\max}}{r} \leq \mathbf{j} \quad \text{iff} \quad n \leq \lfloor \frac{r \cdot \mathbf{j} - b}{L_{\max}} \rfloor$$

Thus, for any path, as long as its hop count is no more than $N = \lfloor (r \cdot \mathbf{j} - b)/L_{\max} \rfloor$, it will meet the delay-jitter bound $\mathbf{j}$. We apply Lemma 1 using the distance function defined in Equation 4 with delay bound $\mathbf{d}$, and hop count restriction N, to get our result. □

Proposition 4 *QoS routing problem* **(D-J)** *is solvable in $E \cdot S$, where E is the number of possible residual bandwidth of links in the network and S is the time for the Bellman-Ford algorithm. Moreover, if several paths are feasible, we can select one with any one of the following properties: minimum delay, minimum delay-jitter, or minimum hop count.*

Proof. Similar to the proof of Proposition 2, we iterate the Bellman-Ford shortest-path algorithm over all possible values of the link residual bandwidth R_k. At each iteration of $r = R_k$, the length function l_r of Equation 5 is used. The only difference is that, as in the proof of Proposition 3, we use the hop count $N_r = \lfloor (r \cdot \mathbf{j} - b)/L_{\max} \rfloor$ to control the number of steps. Also, only those links with $R_i \geq r$ are considered in each step. □

(c) Delay and Buffer Space Constraints

The problem of finding a path satisfying both an end-to-end delay bound and buffer space constraints can be formulated as follows.

Path Finding Problem (D-B-r): Given a delay bound $\mathbf{d}$, a buffer space constraint $\mathbf{b}_u$ for each node u, a leaky bucket $\langle\sigma, b\rangle$, and a bandwidth r ($r \geq \sigma$) to reserve. Find a path $\mathbf{p}$, such that $D(\mathbf{p}, r, b) \leq d$, $B(\mathbf{p}, b, h) \leq \mathbf{b}_h$, and $r \leq R_j (\forall j \in \mathbf{p})$, where R_j is the residual bandwidth of link j and $\mathbf{b}_h$ is the buffer space constraint for the node h hops from the source.

Path Finding Problem (D-B): Given a delay bound $\mathbf{d}$, a buffer space constraint $\mathbf{b}_u$ for each node u, and a leaky bucket $\langle\sigma, b\rangle$. Find a path $\mathbf{p}$ such that $D(\mathbf{p}, r, b) \leq \mathbf{d}$, and $B(\mathbf{p}, b, h) \leq \mathbf{b}_h$, for some r with $\sigma \leq r \leq R_j$ ($\forall j \in \mathbf{p}$), where R_i is the residual bandwidth of link j and $\mathbf{b}_h$ is the buffer space constraint for the node h hops away from the source.

Proposition 5 *QoS routing problem* **(D-B-r)** *is solvable by a modified version of the Bellman-Ford algorithm in $O(m \cdot L)$, where m is the number of nodes and L the number of links in the network. Moreover, if several paths are feasible, we can select one with either the minimum delay or the minimum hop count.*

Proof. For each node u in the network, the buffer space constraint $\mathbf{d}_u$ defines a bound on the hop count $N_u = \lfloor(\mathbf{d}_u - b)/L_{\max}\rfloor$, such that node u cannot appear in a path more than N_u hops away from the source. We define the same length function l as in Equation 4 and apply the Bellman-Ford shortest-path algorithm. During step h, we consider only those nodes with $N_u \geq h$. Finally, we select a path from the result paths of all steps. To conclude the proof, we need to show that if there exists a path P' from s to d that satisfies the delay $\mathbf{d}$ and hop count constraints N_j for all nodes j on the path, the modified Bellman-Ford algorithm must find a path P that has no more hops than P', $l(P) \leq l(P')$, and P satisfies the hop count constraints for all nodes on the path. We use induction on h, the hop count of P'. The results clearly applies for $h = 1$. Assuming the result applies for h, we have to show that it applies for $h + 1$. Let w be the last node on the path P' prior to the destination of P', and Q' the path obtained by removing the last hop from P'. Using the induction hypothesis, there exists a path Q such that Q has no more hops than Q', $l(Q) \leq l(Q')$, Q satisfies the hop count constraints, and Q is the shortest path with no more than the h hops found by the Bellman-Ford algorithm. Since the path concatenating Q and the link from w to d is a possible choice, the Bellman-Ford algorithm will find a path P with at most $(h + 1)$ hops at the $(h + 1)$-th iteration, such that P satisfies hop count constraints and

$$l(P) \leq l(Q) + l(w, d) \leq l(Q') + l(w, d) = l(P'). \qquad \square$$

Proposition 6 *QoS routing problem* **(D-B)** *is solvable in $O(m \cdot E^2)$, where m is the number of nodes and E the number of links in the network. Moreover, if there are feasible paths, we can select one with either the minimum delay or with the minimum hop count.*

Proof. Similar to the proof of Proposition 2, we iterate the modified version of the Bellman-Ford shortest-path algorithm in the proof of Proposition 5 over all possible values of the link residual bandwidth R_k. At each iteration of $r = R_k$ for some link k, the same length function l_r as in the proof of Proposition 2 is used. As in the proof of Proposition 5, for every node u, we use the hop count constraint N_u to achieve the buffer space constraint $\mathbf{b}_u$, and let the Bellman-Ford shortest-path algorithm search only those nodes whose hop count bound are larger than the length of the path currently being considered and only those links with $R_i \geq r$. □

(d) Delay, Delay-Jitter, and Buffer Space Constraints

The problem of finding a path satisfying delay, delay-jitter, and buffer space constraints can be formulated as follows.

Path Finding Problem (D-J-B-r): Given a delay bound $\mathbf{d}$, a delay-jitter bound $\mathbf{j}$, and buffer space constraints $\mathbf{b}_u$ for all the node u, a leaky bucket $\langle \sigma, b \rangle$, and a bandwidth r $(r \geq \sigma)$ to reserve, find a path $\mathbf{P}$, such that $D(\mathbf{p}, r, b) \leq d$, $J(\underline{\mathrm{p}}, r, b) \leq \mathbf{j}$, $B(\mathbf{p}, b, h) \leq \mathbf{b}_h$ for all nodes on the path, and $r \leq R_j$ $(\forall j \in \mathbf{p})$, where R_j is the link residual bandwidth and $\mathbf{b}_h$ is the buffer space constraint for the node with h hops from the source.

Path Finding Problem (D-J-B): Given a delay bound $\mathbf{d}$, a delay-jitter bound $\mathbf{j}$, and buffer space constraints $\mathbf{b}_i$ for all the node i, and a leaky bucket $\langle \sigma, b \rangle$. Find a path $\mathbf{p}$, such that $D(\mathbf{p}, r, b) \leq d$, $J(\underline{\mathrm{p}}, r, b) \leq \mathbf{j}$, and $B(\mathbf{p}, b, h) \leq \mathbf{b}_h$ for all nodes on the path, and for some r with $\sigma \leq r \leq R_j (\forall j \in \mathbf{p})$, where R_i is the link residual bandwidth and $\mathbf{b}_h$ is the buffer space constraint for the node with h hops from the source.

Proposition 7 *QoS routing problem* **(D-J-B-r)** *is solvable by a modified version of the Bellman-Ford algorithm in $O(m \cdot L)$, where m is the number of nodes and L the number of links in the network. Moreover, if there are feasible paths, we can select one with any of the following properties: minimum delay, minimal delay-jitter, or minimum hop count.*

Proof. To meet the delay-jitter bound, we only need to control the number of steps in the algorithm in the proof of Proposition 5. □

Proposition 8 *QoS routing problem* **(D-J-B)** *is solvable in $E \cdot S$, where E is the number of all possible residual bandwidth of links in the network and S the time for the Bellman-Ford algrithm. Moreover, if there are feasible paths, we can select one with any of the following properties: minimum delay, minimum delay-jitter, or minimum hop count.*

Proof. The proof is similar to the proof of Proposition 6, except that the delay-jitter bound is used to limit the number of iterations. □

4 RELATED WORK

For traffic with bandwidth guarantees, many studies have contributed QoS path selection algorithms. Breslau et.al. (1993) use an adaptive load-based routing algorithm. Wang and Crowcroft (1996) suggest shortest-widest path. Gawlick et.al. (1995) propose to use shortest path with exponential cost function for permanent connections. Guerin et.al. (1996) suggest shortest-widest path. The dynamic-alternative path (Section 2) is based on results of dynamic alternative path for telecommunications networks (Gibbens and Kelley, 1988).

For traffic with delay guarantees, several studies propose heuristics to tackle the NP-complete problem (Jaffe, 1984; Salama, et.al. 1997). Rampal (1995) evaluates several path selection algorithms. Wang and Crowcroft (1996) gives a careful study of the complexity of QoS path selection. Przygienda (1995) identifies a subset of path selections that can be done in polynormal time. Rosen et.al. (1991) propose an algorithm that is similar to the algorithm in Proposition 2 in a different setting. Guerin and Orda (1997) study more general QoS path selection problems when the routing information is inaccurate, and notice the algorithm of Proposition 2. Pornavilai et. al (1997) consider the problem of routing traffic with multiple QOS constraints, but they assume that the bandwidth to be reserved is known.

5 CONCLUSION

Quality-of-Service (QoS) routing selects paths that satisfy QoS constraints while achieving high resource efficiency. We study QoS routing for traffic requiring bandwidth and delay guarantees. For traffic with bandwidth guarantee, we present an initial evaluation of several routing algorithms. We show that several routing algorithms that favor paths with fewer hops (widest-shortest, shortest distance, and dynamic alternative path) perform well, while algorithms that favor longer paths (shortest-widest) result in less efficient resource utilization. Selecting paths for traffic with end-to-end delay guarantees typically requires satisfying multiple QoS constraints, which is in general computationally intractable. However, the routing problem can be simplified if there are dependencies between the QoS constraints, as is the case in networks using certain classes of scheduling algorithms. Specifically, we show that for a broad class of WFQ-like scheduling algorithms, finding a path satisfying bandwidth, delay, delay-jitter, and/or buffer space constraints is solvable by a modified version of the Bellman-Ford shortest-path algorithm in polynomial time. The bandwidth to be reserved is selected by the routing algorithm.

REFERENCES

Bertsekas D. and Gallager R. (1987) *Data Networks*, Prentice-Hall, 1987.

Breslau, L. Estrin, D. and Zhang, L. (1993) A Simulation Study of Adaptive Source Routing in Integrated Service Networks. *USC CSD Tech. Rep.*

Bolotin V.A. (1994) Modeling Call Holding Time Distributions for CCS Network Design and Performance Analysis. *IEEE JSAC*, (12)(3), 433–438.

Garey, M.R. and Johnson, D.S. (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman and Company, New York.

Gawlick, R. Kalmanek, C. and Ramakrishnan, K.G. (1995) On-line Routing for Permanent Virtual Circuits. *INFOCOM'95*, Boston, USA.

Gibbens, R.J. Kelley, F.P. and Key P.B. (1988) Dynamic Alternative Routing — Modelling and Behaviour. *Proceedings of the 12th ITC.*

Guerin, R. Orda, A. and Williams, D. (1996) QoS Routing Mechanisms and OSPF Extensions. *Internet Draft*, draft-guerin-qos-routing-ospf-00.txt.

Guerin, R. and Orda, A. (1997) QoS-based Routing in Networks with Inaccurate Information: Theory and Algorithms. *INFOCOM'97*, Japan.

Jaffe, J. (1984) Algorithms for Finding Paths with Multiple Constraints. *Networks*, (1)**14**, 95–116.

Ma, Q. Steenkiste, P. and Zhang, H. (1996) Routing High-Bandwidth Traffic in Max-Min Fair Share Networks. *SIGCOMM'96*, Stanford, 206–217.

Pornavalai, C. Chakraborty G. and Shiratori N. (1997) QoS Based Routing Algorithm in Integrated Services Packet Networks. January 1997.

Rosen, J.B. Sun, S.Z. and Xue, G.L. (1991)ref08 Algorithms for the Quickest Path Problem. *Computers and Operations Research*, 18:579-584.

Salama, H.F. Reeves, D.S. and Viniotis, Y. (1997) A Distributed Algorithm for Delay-Constrainted Unicast Routing. *INFOCOM'97*, Japan.

Wang, Z. and Crowcroft, J. (1996) Quality-of-Service Routing for Supporting Multimedia Applications. *IEEE JSAC*, **14**, 1228–1234.

Zhang, H. (1990) Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks. *Proceedings of the IEEE*, (10)**83**.

6 BIOGRAPHY

Qingming Ma is a PhD candidate in the School of Computer Science at Carnegie Mellon University. His research interests are in QoS provisioning and resource management for integrated-service networks. He received his B.S. from Yangzhou University, his M.Eng. from the Changsha Institute of Technology, China, and his M.S. from Carnegie Mellon University. Information on his research can found on his Home Page at URL http://www.cs.cmu.edu/~qma

Peter Steenkiste is a Senior Research Scientist in the School of Computer Science at Carnegie Mellon University. His research interests are in high-performance networking, end-to-end quality of service, and distributed computing. He received the degree of Electrical Engineer a the University of Ghent, Belgium, and the Masters and PhD degrees from Stanford University. Information on Peter Steenkiste's research can found on his Home Page at URL http://www.cs.cmu.edu/~prs