CCIT Report #495 July 2004

QoS Routing with Efficient Traffic Reshaping and Deadline Allocation in EDF Networks

Erez Biton and Ariel Orda
Department of Electrical Engineering
Technion - Israel Institute of Technology
Haifa 32000, Israel
{berez@tx,ariel@ee}.technion.ac.il

Abstract

We consider QoS routing schemes for connections with end-to-end delay requirements in networks that employ the earliest deadline first (EDF) scheduling discipline. More precisely, we consider the class of rate-controlled EDF, for which traffic is reshaped at each node along the path. Previous EDF routing proposals did not consider the possibility of reshaping the traffic with *different parameters at each hop*. Moreover, the two problems of path selection and assignment of deadlines along the chosen path were considered *independently*. On the other hand, it is well known that, by allowing to reshape traffic with different parameters at each hop, better end-to-end bounds can be obtained. Accordingly, in this study we consider the *joint* problem of identifying a feasible path and optimizing the reshaping parameters along the path. This way, we broaden the space of feasible solutions.

Next, we turn to consider the problem of optimizing the route choice in terms of balancing the loads and accommodating multiple connections. Our first scheme identifies a feasible path (if one exists) with the maximum bottleneck residual rate. Then, in order to achieve better utilization of network resources, the second scheme considers the *joint* problem of route selection and assignment of deadlines along the chosen path. By way of simulations, we demonstrate the advantages of our schemes.

I. Introduction

Emerging broadband high speed networks are expected to support real time and multimedia applications, with various Quality of Service (QoS) requirements. Accordingly, a key issue in the design of broadband architectures is how to provide the resources in order to meet the requirement of each connection, and, moreover, how to meet that goal in a networkwide efficient manner. The establishment of efficient QoS routing schemes is, undoubtedly, one of the major building blocks in such architectures.

One of the major problems in the establishment of a connection with QoS guarantees arises from the need to assign some end-to-end requirements, mainly delay, into local requirements, which would indicate how to reserve resources along the route. Obviously, the identification of a feasible, furthermore "optimal" route for such a connection, greatly depends on our ability to perform such an assignment.

The ability to provide end-to-end delay guarantees depends on the scheduling discipline employed in the network. Such disciplines are characterized by bounds on the maximal delay that any link can incur, and hence a corresponding bound on the end-to-end delay can be derived. Such a bound provides a valuable tool for quantifying the quality of a path in terms of its ability to meet the QoS delay requirement. The corresponding routing problem is, therefore, to identify the path that has the best performance, according to that bound and with respect to the QoS requirement.

For example, Generalized Processor Sharing (GPS) [1], also known as Weighted Fair Queueing (WFQ), is a well known scheduling discipline. Indeed, the QoS routing problem in networks with GPS scheduling has been widely explored (*e.g.*, [2, 3, 4, 5, 6]). Yet, another important discipline is the Earliest Deadline First (EDF) scheduling discipline [7]. EDF has been proven to be an optimal scheduling discipline in the sense that if a set of sessions is schedulable under any scheduling discipline (*i.e.*, if the session's packets can be scheduled in such a way that all of their deadlines are met), then the set is also schedulable under EDF [8]. Also, a certain class of EDF schedulers, namely *Rate-Controlled EDF*, was proven to outperform GPS in providing end-to-end delay guarantees in a network [9]. Consequently, the EDF scheduling discipline has been widely investigated as well. In particular, the establishment of schedulability conditions and efficient admission control schemes have been considered in [10, 11]. Furthermore, the establishment of efficient end-to end bounds based on per-node traffic shaping has been studied in [9, 12]. Nevertheless, the corresponding QoS routing and resource assignment problems have not been fully explored. Some simple routing schemes that aim at finding a feasible path have been proposed [13, 14]. However, those schemes do not aim at optimizing the deadline assignment nor the route selection, in terms of maximizing the ability to accommodate future calls. Consequently, the authors of [15] have proposed new resource division policies (*i.e.*, deadline assignment policies). However, the schemes proposed in [15] consider routing and resource division independently. Furthermore, although it is known that reshaping the traffic with potentially different parameters at each node in the network might reduce the obtained end-to-end delay bound [9], none of the above routing schemes considers such traffic reshaping. In other words, all the proposed routing schemes consider the same traffic parameters along the path, and in particular the same traffic parameters as at the entrance to the network. Clearly, such an assumption results in loose end-to-end delay guarantees and lower network resource utilization (*i.e.*, higher blocking probability). Consequently, under such settings, GPS networks might outperform EDF networks in terms of session blocking probability, as suggested in [16].

Focusing on burstiness constrained traffic (also known as single-leaky-bucket traffic), we consider the *joint* problem of optimizing the traffic reshaping parameters along a path (to obtain lower end-to-end delay bounds) and identifying the *quickest path*, *i.e.*, the path with the minimum end-to-end delay bound. Then, we turn to consider the more complex problem of optimizing the route choice in terms of balancing the loads and accommodating multiple connections.

The rest of the paper is structured as follows. In Section II, we formulate the model. Next, in Section III, we discuss the EDF schedulability conditions and some prerequisite results. In Section IV, we consider the problem of finding feasible paths and in particular quickest paths. More specifically, we establish new routing schemes that identify the quickest path while optimizing the traffic reshaping parameters. Then, in Section V, we turn to consider the problem of optimizing the route choice as well as the deadline assignment. in Section VI, we illustrate the efficiency of our routing schemes through simulations. Finally, in Section VII, we conclude our study.

II. Model Formulation

Given is a network across which sessions need to be routed. The network is represented by a directed graph $G(V, E)$, in which nodes represent switches and arcs represent links. V is the set of nodes and E is the set of interconnecting links; let $|V| = N$ and $|E| = M$.

Each link $l \in E$ is characterized by (i) a service rate R_l and (ii) a *constant delay* value δ_l, related to the link's speed, propagation delay and maximal transfer unit.

We assume that the *Rate-Controlled Earliest Deadline First (RC-EDF)* service discipline is employed in each link $l \in E$. Accordingly, traffic from a particular connection entering a switch passes through a traffic shaper before being delivered to the scheduler. The traffic shaper regulates traffic, so that the output of the shaper satisfies certain pre-specified traffic characteristics. We consider the traffic model introduced in [17], of *burstiness constrained* processes (also known as *single-leaky-bucket* traffic). The traffic shaper reshapes the incoming traffic by delaying packets so that the output is BC, and then delivers them to the scheduler. The EDF scheduler associates a deadline $t + d^i$ with each packet of a session i that arrives at time t. The packets are served in the order of their assigned deadlines. For ease of presentation, we assume a preemptive EDF scheduler (or, alternatively, negligible packet sizes).

We assume a source ("explicit") QoS routing framework, in which link state information is exchanged and maintained up-to-date among network nodes for path computation. Routing decisions are based on the image of the network at the source node.

A session i in the network is characterized by the following parameters:

- Source and destination nodes s^i and t^i, correspondingly.
- A maximal traffic burst σ_0^i.
- A traffic upper rate ρ_0^i.
- A required end-to-end delay bound D^i.

A session should be routed through some path $\mathbf{p}$ between the corresponding source and destination nodes. Let H be the maximal possible number of hops in a path. We denote by $n(\mathbf{p})$ the number of hops (*i.e.*, links) of a path $\mathbf{p}$. We shall also denote by $D^i(\mathbf{p})$ the guaranteed end to end delay to session i along path $\mathbf{p}$.

Denote the set of sessions at link $l \in E$ by $\mathcal{I}_l$ and the number of sessions at link l by I_l. Let $I_{\max}$ be the maximal number of sessions at any link, that is $I_{\max} = \max_{l \in E} I_l$. Also, denote the residual rate of a link l by R'_l, where $R'_l = R_l - \sum_{i \in \mathcal{I}_l} \rho_0^i$. Finally, let $R'_{\max}$ be the maximal residual rate in the network, that is $R'_{\max} = \max_{l \in E} R'_l$.

Without loss of generality, all quantities are normalized, such that the minimum (*i.e.*, basic resolution) unit is "1".

III. EDF Schedulability Conditions

We consider flows that are (σ, ρ) - *burstiness constrained*. More formally, consider the data flow of a session i with the amount of arrivals in the time interval $[t_1, t_2]$ denoted by $A^i[t_1, t_2]$. The flow is (σ^i, ρ^i)-*burstiness constrained* if $A^i[t, t+\tau] \leq \sigma^i + \rho^i, \ \forall \tau > 0$.

Let $\mathcal{I}_l$ be a set of flows entering an EDF scheduler in link l with a service rate R_l. Assume that each session i is characterized by a (σ^i, ρ^i)-burstiness constrained flow and a maximum packet queueing delay of d_l^i. Then,

the set $\mathcal{I}_l$ is *EDF-schedulable* if and only if and only if the following two conditions hold:
(i) the *stability condition*, $\sum_{i\in\mathcal{I}_l}\rho^i < R_l$,
and (ii) the *schedulability condition*,

$$R_l t \geq \sum_{i\in\mathcal{I}_l} 1\left(t-d_l^i\right)\left(\sigma^i+\rho^i\left(t-d_l^i\right)\right), \forall t \geq 0, \tag{1}$$

where $1\left(t\right)=\begin{cases}0 & t<0\\ 1 & otherwise\end{cases}$.

Following [11], we define the *link work availability function* $F_l : [0,\infty) \longrightarrow [0,\infty)$ as

$$F_l\left(t\right) = R_l t - \sum_{i\in\mathcal{I}_l} 1\left(t-d_l^i\right)\left(\sigma^i+\rho^i\left(t-d_l^i\right)\right). \tag{2}$$

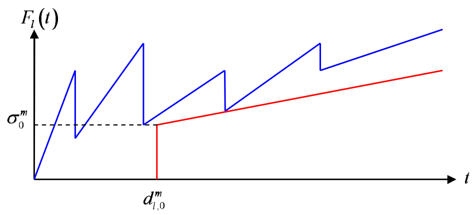


Fig. 1. A typical instance of the work availability function

A typical instance of $F_l\left(t\right)$ is depicted in Figure 1. $F_l\left(t\right)$ specifies the worst case amount of work (in bits) available at time t at the EDF scheduler in link l, while still guaranteeing session i a maximum packet delay of d_l^i, for $1 \leq i \leq I_l$. Clearly, the schedulability condition (1) becomes $F_l\left(t\right) \geq 0\ \forall t \geq 0$. Accordingly, the state of a link $l \in E$ is given by $F_l\left(t\right)$.

It is easy to see that $F_l\left(t\right)$ linearly increases with discontinuities at times $(d_l^i)_{i\in\mathcal{I}_l}$, at which the function decreases in the amount of $(\sigma^i)_{i\in\mathcal{I}_l}$. Note that the local minima of the function $F_l\left(t\right)$ are obtained at times $(d_l^i)_{i\in\mathcal{I}_l}$. Then, the schedulability condition is equivalent to $F_l\left(u\right) \geq 0\ \forall u \in (d_l^i)_{i\in\mathcal{I}_l}\bigcup\{0\}$ [11]. Let us assume, without loss of generality, that the flows in $\mathcal{I}_l = \{1,2,\ldots,I_l\}$ are ordered by (d_l^i): $i<j \Rightarrow d_l^i \leq d_l^j \forall i,j \in \mathcal{I}_l$. Let $d_l^0 = 0$. Denote the local minima values by w_l^i, where $w_l^0 = 0$ and $w_l^i = F_l\left(d_l^i\right), i = 1,2,\ldots,I_l$. Also, denote the slopes of $F_l\left(t\right)$ by r_l^i, where $r_l^i = R_l - \sum\limits_{j\leq i}\rho^i$, $i = 0,1,\ldots,I_l$. Then, the work availability function is fully specified by the set $(d_l^i, w_l^i, r_l^i)_{0\leq i\leq I_l}$. Therefore, the set $(d_l^i, w_l^i, r_l^i)_{0\leq i\leq I_l}$ constitutes the *link parameters*, which specify the state of the link $l \in E$. Obviously, this set should be updated each time a connection is admitted to the scheduler or leaves it. An efficient update algorithm can be found in [11].

Consider a link l and a pending session m with traffic parameters (σ_0^m, ρ_0^m). Then, the minimum queueing delay $d_{l,0}^m$ that can be guaranteed to m can be found by determining the leftmost position of $\sigma_0^m + \rho_0^m \cdot t$ such that it is below the graph of $F_l\left(t\right)$ for all $t \geq 0$, as in Figure 1. An efficient scheme that is given the link and traffic parameters and calculates $d_{l,0}^m$ can be found in [11].

IV. Finding Feasible Paths

In the context of EDF routing, we seek both a path and a deadline allocation at each link along the chosen path. Accordingly, we define a *path-deadline assignment* as follows.

Definition 1: A *path-deadline assignment* is a path $(\mathbf{p}, \underline{d})$, where $\underline{d} = \{d_l\}_{l \in \mathbf{p}}$ is the set of deadline assignments at each link $l \in \mathbf{p}$.

We begin with the basic problem of identifying a feasible path-deadline allocation. If several feasible paths exist, we seek a path with the minimal end-to-end delay. Consider first the standard case, where the traffic at each node is shaped with the parameters at the entrance to the network. In this case, the problem is formulated as follows.

Quickest Feasible Path Problem (QFP): *Given are a network $G(V, E)$, with a service rate R_l, a propagation delay δ_l, and a work availability function $F_l(t)$ for each $l \in E$. Also, given is a session m with source s^m, destination t^m, upper rate ρ_0^m, burst σ_0^m and an end-to-end delay requirement D^m. Find a feasible path-deadline assignment $(\mathbf{p}^m, \underline{d}^m)$,* i.e., *a path between s^m and t^m and a deadline allocation $\{d_l^m\}_{l \in \mathbf{p}^m}$, such that:*

1. $D^m(\mathbf{p}^m) = \sum_{l \in \mathbf{p}^m} (d_l^m + \delta_l) \leq D^m$,

2. for all $l \in \mathbf{p}^m$:

$$F_l(t) - 1(t - d_l^m)(\sigma_0^m + \rho_0^m (t - d_l^m)) > 0 \ \forall t > 0.$$

If there are several such paths, the one with the minimum end-to-end delay bound is selected, i.e., *a path with the minimum $D^m(\mathbf{p}^m)$.*

Distributed routing schemes that solve this problem have been proposed in several studies (*e.g.*, [13, 14, 11, 18]). Those routing schemes are guaranteed to find a feasible path, if one exists. Basically, the schemes employ the following algorithm, which is specified here for completeness.

Algorithm QFP (sketch): For each link $l \in E$, calculate the minimum queueing delay $d_{l,0}^m$ that link l can guarantee to session m, while maintaining

$$F_l(t) - 1\left(t - d_{l,0}^m\right)\left(\sigma_0^m + \rho_0^m\left(t - d_{l,0}^m\right)\right) > 0, \ \forall t > 0.$$

Then, find the shortest path with respect to the metric $\left\{d_{l,0}^m + \delta_l\right\}$. If the sum of the queueing and propagation delays along the identified path $\tilde{\mathbf{p}}$ is at most D^m , *i.e.*, $D^m(\mathbf{p}^m) \leq D^m$, then return $\tilde{\mathbf{p}}$, $d_l^m = d_{l,0}^m$; otherwise, there is no feasible path.

Next, consider the more complex problem of identifying a feasible path while reshaping the traffic at each node along the path. It is known that the selection of the traffic reshaping parameters influences the schedulable region under RC-EDF [9]. In other words, proper selection of the traffic reshaping to be performed at each node may result in a lower end-to-end delay bound. For simplicity, we focus on the restricted case of single-leaky-buckets. Accordingly, we define a *path-reshaping-deadline assignment* as follows.

Definition 2: A path-reshaping-deadline assignment is a path $\left(\mathbf{p}, \underline{(\sigma, \rho)}, \underline{d}\right)$, where $\underline{(\sigma, \rho)} = \{(\sigma_l, \rho_l)\}_{l \in \mathbf{p}}$ are the reshaping parameters at each link $l \in \mathbf{p}$ and $\underline{d} = \{d_l\}_{l \in \mathbf{p}}$ is the deadline assignment at each link $l \in \mathbf{p}$.

Then, the problem is formulated as follows.

Quickest Feasible Path with Traffic Reshaping Problem (QFPTS): *Given are a network $G(V,E)$, with a service rate R_l, a propagation delay δ_l, and a work availability function $F_l(t)$ for each $l \in E$. Also, given is a session m with source s^m, destination t^m, upper rate ρ_0^m, burst σ_0^m and an end-to-end delay requirement D^m. Find a feasible path-reshaping-deadline assignment $\left(\mathbf{p}^m, \underline{(\sigma^m,\rho^m)}, \underline{d}^m\right)$,* i.e., *a path between s^m and t^m, reshaping parameters $\{(\rho_l^m,\sigma_l^m)\}_{l\in\mathbf{p}^m}$, and a deadline allocation $\{d_l^m\}_{l\in\mathbf{p}^m}$, such that:*

1. $\max\limits_{l\in\mathbf{p}^m}\left(\frac{(\sigma_0^m-\sigma_l^m)^+}{\rho_l^m}\right) + \sum\limits_{l\in\mathbf{p}^m}(d_l^m+\delta_l) \le D^m$

2. for all $l \in \mathbf{p}^m$: $\rho_l^m \ge \rho_0^m$,

3. for all $l \in \mathbf{p}^m$:

$$F_l(t) - 1(t-d_l^m)(\sigma_l^m + \rho_l^m(t-d_l^m)) > 0,\ \forall t > 0.$$

If there are several such paths, the one with the minimum end-to-end delay bound is selected, i.e., *a path with the minimum*

$$D^m(\mathbf{p}^m) = \max_{l\in\mathbf{p}^m}\left(\frac{(\sigma_0^m-\sigma_l^m)^+}{\rho_l^m}\right) + \sum_{l\in\mathbf{p}^m}(d_l^m+\delta_l).$$

Note that the maximum reshaping delay (*i.e.*, $\max\limits_{l\in\mathbf{p}^m}\frac{(\sigma_0^m-\sigma_l^m)^+}{\rho_l^m}$) is incurred only once, and is independent of the number of hops on the path [9].

A. Reshaping both the traffic burst as well as the traffic rate

Let $W_l(\sigma_l^m,\rho_l^m)$ be the minimum delay that can be guaranteed to a session m at link l, as a function of the reshaping parameters σ_l^m and ρ_l^m. Accordingly, for $0 \le \sigma_l^m \le \sigma_0^m$ and $\rho_0^m \le \rho_l^m < R_l'$:

$$W_l(\sigma_l^m,\rho_l^m) = \left\{\min d | F_l(t) - 1(t-d)(\sigma_l^m + \rho_l^m(t-d)) > 0 \forall t > 0\right\}.$$

Obviously, we have that $W_l(\sigma_0^m,\rho_0^m) = d_{l,0}^m$. Furthermore, given the session m traffic reshaping parameters at each link $l \in \mathbf{p}$, (σ_l^m,ρ_l^m), the end-to-end delay along a path $\mathbf{p}$ is given by

$$D^m(\mathbf{p}) = \max_{l\in\mathbf{p}}\left(\frac{(\sigma_0^m-\sigma_l^m)^+}{\rho_l^m}\right) + \sum_{l\in\mathbf{p}}(W_l(\sigma_l^m,\rho_l^m)+\delta_l).$$

Suppose that the maximum allowed reshaping delay along a path $\mathbf{p}$ is C time units, that is

$$\max_{l\in\mathbf{p}}\left(\frac{(\sigma_0^m-\sigma_l^m)^+}{\rho_l^m}\right) \le C.$$

Then, at each link l we seek the session m reshaping parameters (σ_l^m,ρ_l^m) that minimize the link's l guaranteed delay, $W_l(\sigma_l^m,\rho_l^m)$. Denote the obtained minimum delay (with a maximum reshaping delay of C time units) by $W_l(C)$. Clearly, $W_l(0) = d_{l,0}^m$. Algorithm Minimum-Link-Delay, specified in Figure 2, identifies the optimum reshaping parameters $(\bar{\sigma}_l^m,\bar{\rho}_l^m)$ and the corresponding minimum delay $W_l(C)$.

Proposition 1: (a) Let $\mathcal{I}_l$ be a schedulable set of sessions at link l, and let $\{w_l^i, d_l^i, r_l^i\}_{0\le i\le I_l}$ be a set of parameters defining the link work availability function. Let (ρ_0^m,σ_0^m) be the traffic parameters of a new session m, at the entrance to the network. Also, let C be the maximal allowed reshaping delay. Then, the

minimum link's guaranteed delay $W_l(C)$, and the corresponding session m reshaping parameters at link l, $(\bar{\sigma}_l^m, \bar{\rho}_l^m)$, are correctly identified by Algorithm Minimum-Link-Delay.
(b) The algorithm's complexity is $O\left(I_l \cdot \log\left(d_{l,0}^m\right)\right)$.

Proof: A maximal reshaping delay C implies that, for all possible reshaping parameters (σ_l^m, ρ_l^m), we have $\sigma_l^m + \rho_l^m C = \sigma_0^m$. Accordingly, a given delay λ can be guaranteed only if $F(\lambda + C) > \sigma_0^m$. Keeping that in mind, the algorithm calculates the maximum rate $r_{\max}$ for which $F_l(t) - \sigma_0^m - r_{\max}(t - \lambda - C) > 0,\ \forall t \geq \lambda + C$. Then, the algorithm calculates the minimum rate $r_{\min}$ for which $F_l(t) - (\sigma_0^m - r_{min}(\lambda + C - t)) > 0$ for all t in the range $\lambda \leq t < \lambda + C$. Accordingly, the algorithm sets ρ_l^m to be such that $\rho_l^m \geq \rho_0^m$ and $r_{\min} \leq \rho_l^m \leq r_{\max}$.

The algorithm considers $O\left(\log\left(d_{l,0}^m\right)\right)$ delay values, and for each delay λ the algorithm involves $O(I_l)$ calculations. Thus the algorithm's complexity is $O\left(I_l \cdot \log\left(d_{l,0}^m\right)\right)$. ■

With the above algorithm at hand, we turn to solve the routing and traffic reshaping problem QFPTS. Algorithm QFPTS, specified in Figure 3, solves this problem. The algorithm executes Dijkstra's shortest path algorithm for all possible values of C. Thus, obviously, the complexity of the algorithm could be pro-

input: $\{w_l^i, d_l^i, r_l^i\}_{0 \leq i \leq I_l}$, (ρ_0^m, σ_0^m), C
output: $(\bar{\sigma}_l^m, \bar{\rho}_l^m)$, $W_l(C)$
1. find $d_{l,0}^m$
2. $H \leftarrow d_{l,0}^m$
3. $L \leftarrow 0$
4. $\lambda \leftarrow d_{l,0}^m$
5. while $\left|\lambda - \frac{L+H}{2}\right| > 1$
   (a) $\lambda \leftarrow \frac{L+H}{2}$
   (b) let j such that $d_l^{j-1} \leq \lambda + C$ and $d_l^j > \lambda + C$
   (c) let b such that $d_l^{b-1} \leq \lambda$ and $d_l^b > \lambda$
   (d) if $w_l^{j-1} + r_l^{j-1}\left(\lambda + C - d_l^{j-1}\right) > \sigma_0^m$
   i. $r_{\max}^1 \leftarrow R_l'$
   ii. $r_{\max}^2 \leftarrow \min\limits_{i \geq j} \frac{w_l^i - \sigma_0^m}{d_l^i - \lambda - C}$
   iii. $r_{\max} \leftarrow \min\left\{r_{\max}^1, r_{\max}^2\right\}$
   iv. $r_{\min}^1 \leftarrow \frac{\sigma_0^m - \left(w_l^{b-1} + r_l^{b-1}\left(\lambda - d_l^{b-1}\right)\right)}{C}$
   v. $r_{\min}^2 \leftarrow \max\limits_{b \leq i \leq j-1} \frac{\sigma_0^m - w_l^i}{\lambda + C - d_l^i}$
   vi. $r_{\min} \leftarrow \max\left\{r_{\min}^1, r_{\min}^2\right\}$
   vii. if $\rho_0^m > r_{\max}$ or $r_{\min} > r_{\max}$ then go to step 5e
   viii. (else) if $\rho_0^m > r_{\min}$ then $\rho_l^m \leftarrow \rho_0^m$
   ix. else $\rho_l^m \leftarrow r_{\min}$
   x. $\sigma_l^m \leftarrow \sigma_0^m - \rho_l^m C$
   xi. $H \leftarrow \lambda$
   (e) else $L \leftarrow \lambda$ (a delay λ cannot be guaranteed)
6. $W_l(C) \leftarrow \lambda$
7. $(\bar{\sigma}_l^m, \bar{\rho}_l^m) \leftarrow (\sigma_l^m, \rho_l^m)$

Fig. 2. Algorithm Minimum-Link-Delay

hibitively large. Therefore, in order to reach an efficient yet computationally tractable solution, we establish an approximation scheme based on quantizing the reshaping delays. For ease of presentation, we begin with an approximation of order 2, *i.e.*, the following algorithm shall derive a solution (path-reshaping-deadline assignment) whose guaranteed end-to-end delay is at most twice greater than that of the quickest.

The session m reshaping delay C could take any value in the interval $\left[0, \frac{\sigma_0^m}{\rho_0^m}\right]$. We group these values into $O\left(\log \frac{\sigma_0^m}{\rho_0^m}\right)$ *reshaping-delay-classes*, such that, for $1 \leq k \leq \lceil \log_2 \frac{\sigma_0^m}{\rho_0^m} \rceil$, reshaping-delay-class k contains all reshaping delays in the range $\left(2^{k-1}, 2^k\right]$.

The Quantized 2-Approximation algorithm, termed Algorithm QFPTS-Q-2, is specified in Figure 4.

Proposition 2: (a) The guaranteed end-to-end delay of the output path $\tilde{\mathbf{p}}^m$ of algorithm QFPTS-Q-2 is at most twice larger then the minimal value, *i.e.*, if $\mathbf{p}^*$ is the quickest path then $D^m\left(\tilde{\mathbf{p}}^m\right) \leq 2 \cdot D^m\left(\mathbf{p}^*\right)$.

(b) The algorithm's complexity is

$$O\left(\log\left(\frac{\sigma_0^m}{\rho_0^m}\right)(N \log N + M)\right).$$

Proof: Let C^* be the reshaping delay of the optimal solution and k^* be the corresponding reshaping-delay-class. Then, by the algorithm, $C^{k^*} \leq 2 \cdot C^*$. Furthermore, since $C^* \leq C^{k^*}$ implies that $W_l\left(C^{k^*}\right) \leq W_l\left(C^*\right)$, it holds that $\sum_{l \in \mathbf{p}(k^*)} W_l\left(C^{k^*}\right) + \delta_l \leq \sum_{l \in \mathbf{p}^*} W_l\left(C^*\right) + \delta_l$. Therefore,

$$C^{k^*} + \sum_{l \in \mathbf{p}(k^*)} W_l\left(C^{k^*}\right) + \delta_l \leq 2 \cdot C^* + \sum_{l \in \mathbf{p}^*} W_l\left(C^*\right) + \delta_l.$$

Then, clearly, $D^m\left(\tilde{\mathbf{p}}^m\right) \leq C^{k^*} + \sum_{l \in \mathbf{p}(k^*)} W_l\left(C^{k^*}\right) + \delta_l$, which implies that $D^m\left(\tilde{\mathbf{p}}^m\right) \leq 2 \cdot D^m\left(\mathbf{p}^*\right)$, and the first part of the proposition follows. The second part of the proposition is straightforward. ■

We proceed to generalize the above algorithm for obtaining an ϵ-approximation. Given a value $\epsilon > 0$, the reshaping delay values are grouped into $O\left(\log_{1+\epsilon} \frac{\sigma_0^m}{\rho_0^m}\right)$ reshaping-delay-classes, such that, for $1 \leq k \leq \lceil \log_{1+\epsilon}\left(\frac{\sigma_0^m}{\rho_0^m}\right) \rceil$, reshaping-delay-class k contains all reshaping delays in the range $\left((1+\epsilon)^{k-1}, (1+\epsilon)^k\right]$. The corresponding Algorithm QFPTS-Q-ϵ is identical to Algorithm QFPTS-Q-2, except that it consists of $O\left(\log_{1+\epsilon} \frac{\sigma_0^m}{\rho_0^m}\right)$ iterations that consider the above $O\left(\log_{1+\epsilon} \frac{\sigma_0^m}{\rho_0^m}\right)$ reshaping-delay-classes.

1. for all possible values of C, $0 \leq C \leq \frac{\sigma_0^m}{\rho_0^m}$
   (a) for each $l \in E$, calculate $W_l(C)$ and $(\bar{\sigma}_l^m, \bar{\rho}_l^m)$ through algorithm Minimum-Link-Delay
   (b) find the shortest path $\mathbf{p}(C)$ w.r.t. $\{W_l(C) + \delta_l\}$ (through Dijkstra's shortest path algorithm)
2. let $\tilde{C}$ such that $\tilde{C} + \sum_{l \in \mathbf{p}(\tilde{C})} \left(W_l\left(\tilde{C}\right) + \delta_l\right)$
is minimized
3. if $\tilde{C} + \sum_{l \in \mathbf{p}(\tilde{C})} \left(W_l\left(\tilde{C}\right) + \delta_l\right) < D^m$
   (a) return $\tilde{\mathbf{p}}^m \leftarrow \mathbf{p}\left(\tilde{C}\right)$, $(\tilde{\sigma}_l^m, \tilde{\rho}_l^m)$, $d_l^m \leftarrow W_l\left(\tilde{C}\right)$
   (b) end
4. else there is no feasible path, end

Fig. 3. Algorithm QFPTS

1. for all $l \in E$, calculate $d_{l,0}^m$
2. $C^0 \leftarrow 0$, $(\bar{\sigma}_l^m, \bar{\rho}_l^m)(0) \leftarrow (\sigma_0^m, \rho_0^m)$
3. find the shortest path $\mathbf{p}(0)$ w.r.t. $D_{l,0}^m + \delta_l$
4. for all k, $1 \leq k \leq \lceil \log_2 \frac{\sigma_0^m}{\rho_0^m} \rceil$
   (a) $C^k \leftarrow 2^k$
   (b) for all $l \in E$, calculate $W_l\left(C^k\right)$ and $(\bar{\sigma}_l^m, \bar{\rho}_l^m)(k)$ through algorithm Minimum-Link-Delay
   (c) find the shortest path $\mathbf{p}(k)$ w.r.t. $\left\{W_l\left(C^k\right) + \delta_l\right\}$ (through Dijkstra's shortest path algorithm)
5. let $\tilde{k}$ such that $C^{\tilde{k}} + \sum_{l \in \mathbf{p}(\tilde{k})} \left(W_l\left(C^{\tilde{k}}\right) + \delta_l\right)$ is minimized
6. if $C^{\tilde{k}} + \sum_{l \in \mathbf{p}(\tilde{k})} \left(W_l\left(C^{\tilde{k}}\right) + \delta_l\right) < D^m$
   (a) return $\tilde{\mathbf{p}}^m \leftarrow \mathbf{p}\left(\tilde{k}\right)$, $(\tilde{\sigma}_l^m, \tilde{\rho}_l^m) \leftarrow (\bar{\sigma}_l^m, \bar{\rho}_l^m)\left(\tilde{k}\right)$, $d_l^m \leftarrow W_l\left(C^{\tilde{k}}\right)$,
   (b) end
7. else there is no feasible path, end

Fig. 4. Algorithm QFPTS-Q-2

Proposition 3: (a) The guaranteed end-to-end delay of the output path $\tilde{\mathbf{p}}^m$ of Algorithm QFPTS-Q-ϵ is at most $1+\epsilon$ larger than the minimum value, *i.e.*, if $\mathbf{p}^*$ is a quickest path, then $D^m(\tilde{\mathbf{p}}^m) \leq (1+\epsilon) \cdot D^m(\mathbf{p}^*)$. (b) The algorithm's complexity is $O\left(\frac{1}{\epsilon}\log\left(\frac{\sigma_0^m}{\rho_0^m}\right)(N\log N + M)\right)$.

Proof: The proof of the first part of the proposition is similar to that of Proposition 2. The second part is immediate from the following relation:

$$O\left(\log_{1+\epsilon}\frac{\sigma_0^m}{\rho_0^m}\right) = O\left(\frac{\log\frac{\sigma_0^m}{\rho_0^m}}{\log(1+\epsilon)}\right) = O\left(\frac{1}{\epsilon}\cdot\log\frac{\sigma_0^m}{\rho_0^m}\right).$$

■

A.1 An Example

We now demonstrate the efficiency of the proposed QFPTS algorithm. Through a simple network example, we indicate that the standard QFP algorithm may not identify a feasible path; in contrast, our QFPTS algorithm is guaranteed to identify a feasible path, if it exists.

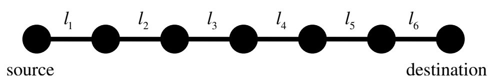


Fig. 5. A 6-hop network example

Consider some source and destination nodes connected by a 6-hop path, as depicted in Fig. 5. The bandwidth of each of the six links is assumed to be 10Mbps. Assume that propagation delays are negligible. Suppose that a session, referred to as *session* 1, with $(1Mb, 2Mbps)$-burstiness constrained traffic, is routed through these links. Also, assume that session 1 is reshaped at each link with the same traffic parameters as at the entrance to the network. Furthermore, assume that the session is assigned a deadline of $200ms$ at each link, which constitute an end-to-end guaranteed delay of $1200ms$. Now, suppose that a new session, referred to as *session* 2, is pending, and suppose that its traffic parameters are $(1Mb, 5Mbps)$, and it requires an end-to-end delay guaranteed of $1000ms$. Under the above setting, the standard QFP algorithm assumes a minimum delay of $200ms$ at each link. A minimum delay of $200ms$ follows since at time $t = 200ms$ the work availability function equals to $1Mb$, that is, $F_l(200ms) = 10Mbps \cdot 200ms - 1Mb = 1Mb$. Accordingly, the minimum end-to-end delay along the path is $1200ms$, which is not feasible. In contrast, our QFPTS algorithm identifies an optimal reshaping at each node; that is, the traffic is reshaped with the parameters $(0Mb, 5Mbps)$, which corresponds to a shaping delay of $500ms$. Consequently, a minimum queueing delay of $0ms$ can be guaranteed at each link. A $0ms$ delay follows since $F_l(t) - 5Mbps \cdot t \geq 0, \ \forall t > 0$. In particular, at time $t = 200ms$, $F_l(t) - 5Mbps \cdot t = 1Mb - 5Mbps \cdot 200ms = 0$. Thus, the path is feasible for session 2 with a guaranteed end-to-end delay of $500ms$. Alternatively, algorithm QFPTS-Q-2 identifies a guaranteed end-to-end delay of at most $1000ms$, which is still feasible.

B. Reshaping only the traffic burst

Here, we consider the special case in which only the maximal burst size of the traffic can be reshaped along the path.

Suppose that, at each link $l \in \mathbf{p}$, the traffic of a session m is shaped with a fixed rate ρ_0^m, as at the entrance to the network. Furthermore, suppose that, at each link l, the traffic burst of session m is reshaped with a parameter σ_l^m. Accordingly, at each link l, the traffic of session m entering the EDF scheduler is (ρ_0^m, σ_l^m)-burstiness constrained.

Define the *link-delay-burst function* $W_l(\sigma)$, for all $0 \leq \sigma \leq \sigma_0^m$, as

$$W_l(\sigma_l^m) = \{\min d | F_l(t) - 1(t-d)(\sigma_l^m + \rho_0^m(t-d)) > 0, \ \forall t > 0\}.$$

$W_l(\sigma_l^m)$ is the minimum delay that can be guaranteed to a session m at link l, as a function of σ_l^m. A typical instance of $W_l(\sigma_l^m)$ (which corresponds to $F_l(t)$ of Figure 1) is depicted in Figure 6. We note that $W_l(\sigma_l^m)$

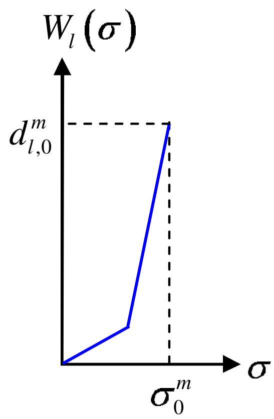


Fig. 6. A typical instance of the link-delay-burst function $W_l(\sigma_l^m)$

is an increasing piecewise linear function of σ_l^m. Furthermore, the number of points at which the slope of the function $W_l(\sigma_l^m)$ changes is less than $2I_l^-$, where I_l^- is the number of sessions at link l with a deadline of at most $d_{l,0}^m$. Denote the burst size instances for which the slope of $W_l\left(\sigma^m\right)$ changes by x_l^j and let $x_l^0 = 0$. Additionally, denote the slope values and the function values of $W_l\left(\sigma^m\right)$ at these points (starting with the right hand side) by a_l^j and y_l^j, respectively. Let J_l be the number of points at which the slope changes, where $J_l \leq 2I_l^-$.

Algorithm Calculate-Link-Delay-Burst, specified in Figure 7, calculates the set $\left(y_l^j, x_l^j, a_l^j\right)_{0\leq j\leq J}$ for a link $l \in E$, given the link parameters $\left(w_l^i, d_l^i, r_l^i\right)_{0\leq i\leq I_l}$ (the link's work availability function) and given the pending session's parameters $\left(\rho_0^m, \sigma_0^m\right)$.

Proposition 4: (a) Let $\mathcal{I}_l$ be a schedulable set of sessions at link l, and let $\left\{w_l^i, d_l^i, r_l^i\right\}_{0\leq i\leq I_l}$ be a set of parameters defining the link work availability function. Also, let $\left(\rho_0^m, \sigma_0^m\right)$ characterize a new session m such that the stability condition $\sum\limits_{i\in\mathcal{I}_l} \rho_l^i + \rho_l^m < R_l$ is satisfied. Then, the link-delay-burst $W_l\left(\sigma^m\right)$, is fully specified by the set $\left\{y_l^j, x_l^j, a_l^j\right\}_{0\leq j\leq J_l}$, obtained by Algorithm Calculate-Link-Delay-Burst.
(b) The algorithm's complexity is $O\left(I_l^2\right)$.

Proof: First, the algorithm finds the minimum guaranteed delay for a zero burst as follows.
Let $\left(\alpha_i\right)_{i\in I_l}$ such that

$$F_l\left(d_l^i\right) = \rho_0^m\left(d_l^i - \alpha_i\right) \quad i \in I_l$$

and let $c_\alpha = \max\left(\alpha_i, 0\right)$.
Let $c_\beta = \max\limits_{i\in I_l; F_l\left(d_l^i\right)=0}\left(d_l^i\right)$ and $c = \max\left(c_\alpha, c_\beta\right)$.
Let b, $1 \leq b \leq I_l + 1$, such that $d_l^{b-1} \leq m < d_l^b$, where $d_l^0 = 0$ and $d_l^{I_l+1} = \infty$.

If $F_l\left(d_l^{b-1}\right) = 0$ then, the minimum guaranteed delay is d_l^{b-1}, *i.e.*, $y_l^0 = d_l^{b-1}$. The slope of $W_l\left(\sigma^m\right)$ is $\frac{1}{r_l^{b-1}}$. Furthermore, let

$$e = \arg\min_{k>b-1} F_l\left(d_l^k\right) - \rho_0^m\left(d_l^k - d_l^{b-1}\right)$$

and let α be such that

$$F_l\left(d_l^e\right) - \rho_0^m\left(d_l^e - \alpha\right) = F_l\left(d_l^{b-1}\right) + r_l^{b-1}\left(\alpha - d_l^{b-1}\right).$$

Then, the slope of $W_l\left(\sigma^m\right)$ changes at $W_l\left(\sigma^m\right) = \alpha$ to $\frac{1}{\rho_0^m}$.

Otherwise (*i.e.*, $F_l\left(d_l^{b-1}\right) \neq 0$), the minimum guaranteed delay for a zero burst is c_α and the first slope of $W_l\left(\sigma^m\right)$ is $\frac{1}{\rho_0^m}$.

Next, the algorithm iterates over the delays $\left\{d_l^b\right\}_{b\leq I_l+1}$ in order to find the corresponding points at which the slope of $W_l\left(\sigma^m\right)$ changes.
Let

$$e = \arg\min_{k>b-1} F_l\left(d_l^k\right) - \rho_0^m\left(d_l^k - d_l^{b-1}\right).$$

If

$$F_l\left(d_l^e\right) - \rho_0^m\left(d_l^e - d_l^{b-1}\right) > F_l\left(d_l^{b-1}\right)$$

then the slope of $W_l\left(\sigma^m\right)$ changes at $W_l\left(\sigma^m\right) = d_l^{b-1}$ to $\frac{1}{r_l^{b-1}}$. Furthermore, the slope of $W_l\left(\sigma^m\right)$ changes again at $W_l\left(\sigma^m\right) = c$ where c is such that

$$F_l\left(d_l^e\right) - \rho_0^m\left(d_l^e - c\right) = F_l\left(d_l^{b-1}\right) + r_l^{b-1}\left(c - d_l^{b-1}\right).$$

input: $\{w_l^i, d_l^i, r_l^i\}_{0\le i\le I_l}$, (ρ_0^m, σ_0^m)
output: $\left\{y_l^j, x_l^j, a_l^j\right\}_{0\le j\le J_l}$

1. $c_\alpha \leftarrow 0; c_\beta \leftarrow 0$
2. for i=1 to I_l do
   - (a) if $w_l^i > 0$ then
     - i. $\alpha_i \leftarrow d_l^i - \frac{w_l^i}{\rho_0^m}$
     - ii. $c_\alpha \leftarrow \max(c_\alpha, \alpha_i)$
   - (b) else $c_\beta \leftarrow \max\left(c_\beta, d_l^i\right)$
3. $c \leftarrow \max(c_\alpha, c_\beta)$
4. $b \leftarrow 0$
5. while $(b \le I_l + 1)$ and $\left(c \ge d_l^b\right)$ do
   - (a) $b \leftarrow b + 1$
6. $j \leftarrow 0$
7. if $w_{b-1} = 0$
   - (a) $y_l^j \leftarrow d_l^{b-1}; x_l^j \leftarrow 0; a_l^j \leftarrow \frac{1}{r_l^{b-1}}$
   - (b) $c \leftarrow w_l^{b-1}; e \leftarrow b - 1$
   - (c) for $k = b$ to $I_l + 1$ do
     - i. if $w_l^k - \rho_0^m\left(d_l^k - d_l^{b-1}\right) < c$
     - ii. $c \leftarrow w_l^k - \rho_0^m\left(d_l^k - d_l^{b-1}\right); e \leftarrow k$
   - (d) $c \leftarrow \frac{w_l^e - w_l^{b-1} + r_l^{b-1} d_l^{b-1} - \rho_0^m d_l^e}{r_l^{b-1} - \rho_0^m}$
   - (e) $j \leftarrow j + 1$
   - (f) $y_l^j \leftarrow c; x_l^j \leftarrow x_l^{j-1} + \frac{y_l^j - y_l^{j-1}}{a_l^{j-1}}; a_l^j \leftarrow \frac{1}{\rho_0^m}$
8. else $y_l^j \leftarrow c_\alpha; x_l^j \leftarrow 0; a_l^j \leftarrow \frac{1}{\rho_0^m}$
9. $b \leftarrow b + 1$
10. while $b < I_l + 1$ and $x_l^j < \sigma_0^m$
    - (a) $c \leftarrow w_l^{b-1}; e \leftarrow b - 1$
    - (b) for $k = b$ to $I_l + 1$ do
      - i. if $w_l^k - \rho_0^m\left(d_l^k - d_l^{b-1}\right) < c$
      - ii. $c \leftarrow w_l^k - \rho_0^m\left(d_l^k - d_l^{b-1}\right); e \leftarrow k$
    - (c) if $w_l^e - \rho_o^m\left(d_l^e - d_l^{b-1}\right) > w_l^{b-1}$ do
      - i. $j \leftarrow j + 1$
      - ii. $y_l^j \leftarrow d_l^{b-1}; x_l^j \leftarrow x_l^{j-1} + \frac{y_l^j - y_l^{j-1}}{a_l^{j-1}}; a_l^j \leftarrow \frac{1}{r_l^{b-1}}$
      - iii. $c \leftarrow \frac{w_l^e - w_l^{b-1} + r_l^{b-1} d_l^{b-1} - \rho_0^m d_l^e}{r_l^{b-1} - \rho_0^m}$
      - iv. $j \leftarrow j + 1$
      - v. $y_l^j \leftarrow c; x_l^j \leftarrow x_l^{j-1} + \frac{y_l^j - y_l^{j-1}}{a_l^{j-1}}; a_l^j \leftarrow \frac{1}{\rho_0^m}$
    - (d) $b \leftarrow b + 1$
11. if $b = I_l + 1$ and $b \ne 1$
    - (a) $j \leftarrow j + 1$
    - (b) $y_l^j \leftarrow d_l^{b-1}; x_l^j \leftarrow x_l^{j-1} + \frac{y_l^j - y_l^{j-1}}{a_l^{j-1}}; a_l^j \leftarrow \frac{1}{r_l^{b-1}}$
12. $J_l \leftarrow j$

Fig. 7. Algorithm Calculate-Link-Delay-Burst

Here, the slope changes to $\frac{1}{\rho_0^m}$.

For each i, $1 \leq i \leq I_l$, the algorithm consists $O(I_l)$ operations. Thus, the algorithm's complexity is $O(I_l^2)$. ■

Now, we have that the end-to-end delay along a path $\mathbf{p}$ is given by

$$D^m(\mathbf{p}) = \max_{l \in \mathbf{p}} \left(\frac{(\sigma_0^m - \sigma_l^m)^+}{\rho_0^m} \right) + \sum_{l \in \mathbf{p}} (W_l(\sigma_l^m) + \delta_l).$$

Corollary 1: Suppose that the session m guaranteed end-to-end delay $D^m(\mathbf{p})$ is minimized by the set $\{\bar{\sigma}_l^m\}_{l \in \mathbf{p}}$. Then, $\{\bar{\sigma}_l^m\}_{l \in \mathbf{p}}$ have the same value, *i.e.*, $\bar{\sigma}_l^m = \bar{\sigma}^m \ \forall l \in \mathbf{p}$.

Proof: By contradiction. Assume that $\bar{\sigma}_l^m = \bar{\sigma}^m \ \forall l \in \mathbf{p}$ does not hold. Then, let $\tilde{\sigma}^m = \min_{l \in \mathbf{p}} \bar{\sigma}_l^m$. Clearly, $\max_{l \in \mathbf{p}} \left(\frac{(\sigma_0^m - \bar{\sigma}_l^m)^+}{r^m} \right) = \max_{l \in \mathbf{p}} \left(\frac{(\sigma_0^m - \tilde{\sigma}^m)^+}{r^m} \right)$. However, since $W_l(\sigma_l^m)$ is increasing in σ_l^m, we have that $W_l(\tilde{\sigma}^m) \leq W_l(\bar{\sigma}_l^m) \ \ \forall l \in \mathbf{p}$. Accordingly, the end-to-end delay along the path $\mathbf{p}$ with a maximal burst size of $\tilde{\sigma}^m \ \forall l \in \mathbf{p}$ is lower than the end-to-end delay with maximal burst sizes of $\bar{\sigma}_l^m \ \forall l \in \mathbf{p}$, which contradicts the optimality of $\{\bar{\sigma}_l^m\}_{l \in \mathbf{p}}$. ■

Thus, aiming at minimizing the end-to-end delay we seek a single maximal burst size value for all the links along the path.

Denote $W(\sigma^m) = \sum_{l \in \mathbf{p}} W_l(\sigma^m)$. It is easy to see that $W(\sigma^m)$ is increasing and piecewise linear. Furthermore, if the slope of $W(\sigma^m)$ changes at σ_0 then for at least one link $l \in \mathbf{p}$, say l_0, the slope of $W_{l_0}(\sigma^m)$ changes at σ_0. Accordingly, the slope of $W(\sigma^m)$ changes at most $2hI_{\max}^-$ times, where $I_{\max}^- = \max_{l \in \mathbf{p}} I_l^-$ and h is the number of hops along $\mathbf{p}$.

Given a path $\mathbf{p}$, the following corollary considers the optimal reshaping burst $\bar{\sigma}^m$ that minimizes the guaranteed end-to-end delay $D^m(\mathbf{p})$.

Corollary 2: Suppose that the session m guaranteed end-to-end delay $D^m(\mathbf{p})$ is minimized by a maximal burst size $\bar{\sigma}^m$. Then, $W(\sigma^m)$ changes its slope at $\bar{\sigma}^m$.

Proof: By contradiction. Assume that $W(\sigma^m)$ is differentiable at $\bar{\sigma}^m$. Then, consider the following two possible cases.

Case 1: $W'(\bar{\sigma}^m) < \frac{1}{\rho_0^m}$.

Then, consider a maximal burst size $\tilde{\sigma}^m = \bar{\sigma}^m + \epsilon$. We have that

$$\frac{\sigma_0^m - \tilde{\sigma}^m}{\rho_0^m} + W(\tilde{\sigma}^m) + \sum_{l \in \mathbf{p}} \delta_l < \frac{\sigma_0^m - \bar{\sigma}^m}{\rho_0^m} + W(\bar{\sigma}^m) + \sum_{l \in \mathbf{p}} \delta_l,$$

which contradicts the optimality of $\bar{\sigma}^m$.

Case 2: $W'(\bar{\sigma}^m) > \frac{1}{\rho_0^m}$.

Then, consider a maximal burst size $\tilde{\sigma}^m = \bar{\sigma}^m - \epsilon$. We have that

$$\frac{\sigma_0^m - \tilde{\sigma}^m}{\rho_0^m} + W(\tilde{\sigma}^m) + \sum_{l \in \mathbf{p}} \delta_l < \frac{\sigma_0^m - \bar{\sigma}^m}{\rho_0^m} + W(\bar{\sigma}^m) + \sum_{l \in \mathbf{p}} \delta_l,$$

which contradicts the optimality of $\bar{\sigma}^m$. ■

Corollary 2 suggests that an optimal routing scheme may limit itself to consider reshaping burst sizes at which $W_l(\sigma^m)$, $\forall l \in E$, change their slopes. Accordingly, for each such burst size, Algorithm *QFPTS-Fixed Rate* (QFPTS-FR), depicted in Figure 8, finds the shortest path w.r.t. $W_l(\sigma^m) + \delta_l$. Then, among the $O(MI_{\max})$ paths, it chooses a path $\tilde{\mathbf{p}}$ and burst $\tilde{\sigma}^m$ that minimize the guaranteed end-to-end delay.

1. for all $l \in E$, call Calculate-Link-Delay-Burst
2. for all values of σ^m, $\sigma^m \in \left\{ \left\{ x_l^j \right\}_{0 \le j \le J_l} \right\}_{l \in E}$
   (a) for all $l \in E$, calculate $W_l(\sigma^m)$
   (b) find the shortest path $\tilde{\mathbf{p}}$ between s^m and t^m w.r.t. $\{W_l(\sigma^m) + \delta_l\}$ (through Dijkstra's shortest path algorithm)
3. among the $O(MI_{\max})$ paths, choose a path $\tilde{\mathbf{p}}$ and burst $\tilde{\sigma}^m$ with the smallest guaranteed end-to-end delay $D^m(\mathbf{p}) = \frac{(\sigma_0^m - \sigma^m)^+}{\rho_0^m} + \sum_{l \in \mathbf{p}} (W_l(\sigma^m) + \delta_l)$.
4. if $D^m(\tilde{\mathbf{p}}) < D^m$ then
   (a) $d_l^m \leftarrow W_l(\tilde{\sigma}^m)$, $\forall l \in \tilde{\mathbf{p}}$
   (b) $(\tilde{\sigma}_l^m, \tilde{\rho}_l^m) \leftarrow (\tilde{\sigma}^m, \rho_0^m)$, $\forall l \in \tilde{\mathbf{p}}$
   (c) return $\tilde{\mathbf{p}}$, $\{d_l^m\}_{l \in \tilde{\mathbf{p}}}$, $\{(\tilde{\sigma}_l^m, \tilde{\rho}_l^m)\}_{l \in \tilde{\mathbf{p}}}$
5. else
   (a) there is no feasible path
   (b) end

Fig. 8. Algorithm QFPTS-FR

Proposition 5: Algorithm QFPTS-FR correctly solves QFPTS problem for a fixed reshaping traffic rate ρ_o^m. The algorithm's complexity is $O(MI_{\max}(N \log N + M))$.

Proof: The first part of the proposition follows from Corollaries 1 and 2. The algorithm consists of $O(MI_{\max})$ executions of Dijkstra's shortest path algorithm. The second part of the proposition immediately follows. ■

While the computational complexity of Algorithm QFPTS-FR is polynomial, it could still be prohibitively large. Therefore, in order to obtain an efficient yet computationally tractable solution, we establish an ϵ-optimal solution with a complexity of $O\left(M(N \log N + M) \frac{1}{\epsilon} \log D_{\max}\right)$. To that end, we quantize the delay deadlines at each link.

Suppose that the allowed values of guaranteed link delay are restricted to a set of $O\left(\log_{1+\epsilon} D_{\max}\right)$ *delay-classes*, where $D_{\max}$ is the maximal end-to-end delay requirement. More precisely, at each link $l \in E$, the delay deadline assignment has to assume a value out of the set $0, 1+\epsilon, (1+\epsilon)^2, \ldots (1+\epsilon)^K$, where $K = \lceil log_{1+\epsilon} D_{max} \rceil$. Accordingly, the work availability function at each link l might change its slope only at the points $0, 1+\epsilon, (1+\epsilon)^2, \ldots (1+\epsilon)^K$. Consequently, the slope of the link-delay-burst function changes at most $2 * K$ times. The corresponding algorithm *QFPTS-FR- Delay Quntized* - ϵ (QFPTS-FR-DQ-ϵ), depicted in Figure 9, is similar to algorithm QFPTS-FR, only that it consists of $O\left(M \log_{1+\epsilon} D_{\max}\right)$ iterations and it assigns deadlines out of the allowed set.

Proposition 6: (a) The guaranteed end-to-end delay of the output path $\tilde{\mathbf{p}}^m$ of algorithm QFPTS-FR-DQ-ϵ is at most $1+\epsilon$ larger then the minimal value obtained by Algorithm QFPTS-FR, *i.e.*, if $\mathbf{p}^*$ is the output path of algorithm QFPTS-FR then $D^m(\tilde{\mathbf{p}}^m) \le (1+\epsilon) \cdot D^m(\mathbf{p}^*)$.

(b) The algorithm's complexity is $O\left(M\left(N\log N+M\right)\frac{1}{\epsilon}\log D_{\max}\right)$.

Proof: For each link $l\in\tilde{\mathbf{p}}$ d_l^m is at most $(1+\epsilon)\,W_l\left(\tilde{\sigma}^m\right)$, *i.e.*, $d_l^m\le(1+\epsilon)\,W_l\left(\tilde{\sigma}^m\right)$. Accordingly,

$$D^m\left(\tilde{\mathbf{p}}\right)=\frac{\left(\sigma_0^m-\sigma^m\right)^+}{\rho_0^m}+\sum_{l\in\mathbf{p}}\left(d_l^m+\delta_l\right)\le(1+\epsilon)\left(\frac{\left(\sigma_0^m-\sigma^m\right)^+}{\rho_0^m}+\sum_{l\in\mathbf{p}}\left(W_l\left(\tilde{\sigma}^m\right)+\delta_l\right)\right)=(1+\epsilon)\,D^m\left(\mathbf{p}^*\right).$$

The algorithm consists of $O\left(M\log_{1+\epsilon}D_{\max}\right)$ executions of Dijkstra's shortest path algorithm. The second part of the proposition is follows from the following relation:

$$O\left(\log_{1+\epsilon}D_{\max}\right)=O\left(\frac{\log D_{\max}}{\log\left(1+\epsilon\right)}\right)=O\left(\frac{1}{\epsilon}\cdot\log D_{\max}\right).$$

■

V. Optimizing the Path Selection

The ability to identify a feasible path for a connection does not yield yet a satisfactory QoS routing solution. Indeed, in order to supervise multiple connections throughout the network, the routing algorithm must consider the efficient use of the consumed resource. There does not seem to be a precise definition for "optimality" of a path in this context, yet it is clear that an efficient scheme should aim at balancing the load throughout the network. In the following, we devise two criteria for balancing the load, as well as the corresponding routing schemes.

A. Rate Consumption Criterion

Consider first the simple criterion of choosing, for a connection request m, a path $\mathbf{p}$ for which the *residual rate* (after establishing the new connection) of its bottleneck link is maximal. That is, we aim at solving the following problem.

1. for all $l\in E$, call Calculate-Link-Delay-Burst
2. for all values of σ^m, $\sigma^m\in\left\{\left\{x_l^j\right\}_{0\le j\le J_l}\right\}_{l\in E}$
   (a) for all $l\in E$, calculate $W_l\left(\sigma^m\right)$
   (b) find the shortest path $\tilde{\mathbf{p}}$ between s^m and t^m w.r.t. $\left\{W_l\left(\sigma^m\right)+\delta_l\right\}$ (through Dijkstra's shortest path algorithm)
3. among the $O\left(MI_{\max}\right)$ paths, choose a path $\tilde{\mathbf{p}}$ and burst $\tilde{\sigma}^m$ with the smallest guaranteed end-to-end delay $D^m\left(\mathbf{p}\right)=\frac{\left(\sigma_0^m-\sigma^m\right)^+}{\rho_0^m}+\sum_{l\in\mathbf{p}}\left(W_l\left(\sigma^m\right)+\delta_l\right)$.
4. if $D^m\left(\tilde{\mathbf{p}}\right)<D^m$ then
   (a) for each $l\in\tilde{\mathbf{p}}$
   i. let k_l such that $W_l\left(\tilde{\sigma}^m\right)>(1+\epsilon)^{k_l-1}$ and $W_l\left(\tilde{\sigma}^m\right)<(1+\epsilon)^{k_l}$
   ii. $d_l^m\leftarrow(1+\epsilon)^{k_l}$
   iii. $\left(\tilde{\sigma}_l^m,\tilde{\rho}_l^m\right)\leftarrow\left(\tilde{\sigma}^m,\rho_0^m\right)$
   (b) return $\tilde{\mathbf{p}}$, $\left\{d_l^m\right\}_{l\in\tilde{\mathbf{p}}}$, $\left\{\left(\tilde{\sigma}_l^m,\tilde{\rho}_l^m\right)\right\}_{l\in\tilde{\mathbf{p}}}$
5. else
   (a) there is no feasible path
   (b) end

Fig. 9. Algorithm QFPTS-FR-RQ-ϵ

Maximum Residual Bottleneck(MRB): *Given are a network $G(V, E)$, with a service rate R_l, a propagation delay δ_l, and a work availability function $F_l(t)$ for each $l \in E$. Also, given is a session m with source s^m, destination t^m, upper rate ρ_0^m, burst σ_0^m and an end-to-end delay requirement D^m. Find a feasible path-deadline assignment $(\mathbf{p}^m, d^m)$,* i.e., *a path between s^m and t^m and a deadline allocation $(d_l^m)_{l \in \mathbf{p}^m}$, such that:*

1. $\sum_{l \in \mathbf{p}^m} (d_l^m + \delta_l) \leq D^m$,

2. for all $l \in \mathbf{p}^m$:

$$F_l(t) - 1(t - d_l^m)(\sigma_0^m + \rho_0^m(t - d_l^m)) > 0 \ \forall t > 0.$$

If there are several such paths, select the one with the maximum residual rate of its bottleneck, i.e., $\max_{\mathbf{p}} \min_{l \in \mathbf{p}} (R'_l - \rho_0^m)$.

Let $R^1, R^2, \ldots R^K$ be the set of all residual rate values, where $K \leq M$. Assume, without the loss of generality, that $R^1 > R^2 > \ldots > R^K$. Algorithm MRB, specified in Figure 10, solves the above problem.

1. for all $l \in E$, calculate the minimum delay $d_{l,0}^m$ that can be guaranteed to session m (through Algorithm MINIMUM-DELAY of [19])
2. find the shortest path $\tilde{\mathbf{p}}$ between s^m and t^m w.r.t. $\left\{d_{l,0}^m + \delta_l\right\}$ (through Dijkstra's shortest path algorithm)
3. if $\sum_{l \in \tilde{\mathbf{p}}} \left(d_{l,0}^m + \delta_l\right) < D^m$ then
   (a) $L \leftarrow 1$, $H \leftarrow K$, $k \leftarrow 1$
   (b) while $k \neq \lfloor \frac{L+H}{2} \rfloor$
   i. $k \leftarrow \lfloor \frac{L+H}{2} \rfloor$
   ii. delete all links l with $R'_l < R^k$
   iii. find the shortest path $\tilde{\mathbf{p}}$ w.r.t. $\left\{d_{l,0}^m + \delta_l\right\}$
   iv. if $\sum_{l \in \tilde{\mathbf{p}}} \left(d_{l,0}^m + \delta_l\right) < D^m$ then $H \leftarrow k$ else $L \leftarrow k$
   (c) $\eta \leftarrow D - \sum_{l \in \tilde{\mathbf{p}}} \left(d_{l,0}^m + \delta_l\right)$
   (d) $d_l^m \leftarrow d_{l,0}^m + \frac{\eta}{h}$, where h is the number of hops along the path $\tilde{\mathbf{p}}$
   (e) return $\tilde{\mathbf{p}}$, $\{d_l^m\}_{l \in \tilde{\mathbf{p}}}$
4. else there is no feasible path, end

Fig. 10. Algorithm MRB

Proposition 7: Algorithm MRB correctly identifies a feasible path (if one such exists) with the maximum residual rate of its bottleneck link. The algorithm's complexity is $O(MI_{\max} + \log K(N \log N + M))$.

Proof: The first part of the proposition is straightforward.
The algorithm calculates MINIMUM-DELAY for all links, which involves $O(MI_{\max})$ operations. Then, the algorithm consists $O(\log K)$ executions of Dijkstra's shortest path algorithm, which involves $O(\log K(N \log N + M))$ operations. ■

B. Delay-Rate Consumption Criterion

The way that a session affects the availability of resources at a link depends on the residual rate as well as the assigned deadline at that link. In other words, a session m that is routed through a link l consumes more resources when it is assigned a smaller deadline. While algorithm MRB balances the rate consumption, the path selection does not account for the allocation of the delay deadlines. More precisely, the MRB scheme balances the deadline assignment along the path $\tilde{\mathbf{p}}$ only after the path is selected. Accordingly, we now consider an alternative routing scheme, which aims at balancing the load as well as optimizing the deadline allocation. To that end, a new criterion for load balancing is called for.

More specifically, we seek a measure for the residual resource available for future connections after admitting a connection. Such a measure should account for the rate consumption as well as the deadline allocation.

Recall that the EDF schedulability condition at each link $l \in E$ is given by the inequality $F_l(t) > 0$, $\forall t > 0$. Thus, when admitting a new connection m with a deadline d_l^m, the following inequality should hold: $\min_{t>d_l^m} \left(F_l(t) - 1(t-d_l^m)\left(\sigma_0^m + \rho_0^m(t-d_l^m)\right)\right) > 0$. Accordingly, for all $d > d_{l,0}^m$, define the *residual resource* function as follows:

$$S_l(d) = \min_{t>d} \left(F_l(t) - 1(t-d)\left(\sigma_l^m + \rho_l^m(t-d)\right)\right)$$

$S_l(d)$ estimates the residual resources available for future connections as a function of the deadline assignment for the newly established connection. In other words, consider a connection established at two links l_1 and l_2 with deadlines d_{l_1} and d_{l_2}, and suppose that $S_{l_1}(d_{l_1}) < S_{l_2}(d_{l_2})$; then, the connection establishment has a more severe impact on link l_1 than on link l_2 in terms of reducing the ability to admit future connection requests. Intuitively, $S_l(d)$ can be found by sliding the function $1(t-d)\left(\sigma_0^m + \rho_0^m(t-d)\right)$ (*i.e.*, increasing

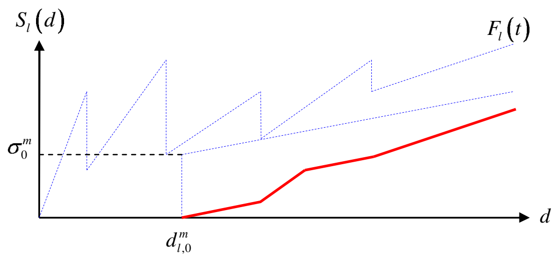


Fig. 11. A typical instance of the residual resource function

d) from its leftmost position $d_{l,0}^m$ to the right (see Figure 11). For each point (*i.e.*, for each d), the value of $S_l(d)$ is given by the minimum distance between $F_l(t)$ and $1(t-d)\left(\sigma_0^m + \rho_0^m(t-d)\right)$. A typical instance of $S_l(d)$ is depicted in Figure 11. Note that $S_l(d)$ is increasing and piecewise-linear with alternating slopes. Denote the delay instances for which the slope of $S_l(d)$ changes by x_l^j and let $x_l^0 = d_{l,0}^m$. Additionally, denote the slope values and the function values of $S_l(d)$ at these points (starting with the right hand side) by a_l^j and y_l^j, respectively. Let J be the number of points at which the slope changes. Then, the residual-resource function is given by the set $\left(y_l^j, x_l^j, a_l^j\right)_{0\le j\le J}$.

Algorithm Calculate-Residual-Resource, specified in Figure 12, calculates the set $\left(y_l^j, x_l^j, a_l^j\right)_{0\le j\le J}$ for a link $l \in E$, given the link parameters $(w_l^i, d_l^i, r_l^i)_{0\le i\le I_l}$ (the link's work availability function) and given the pending session's parameters (ρ_0^m, σ_0^m). First, the algorithm considers a zero traffic burst. Then, at step 6, the algorithm makes the necessary modifications to account for a traffic burst of σ_0^m. Assuming a zero burst, the algorithm considers all the minima of the work availability function $F_l(t)$. For each minimum, it identifies the points at which the slope changes. More specifically, suppose there is a minimum at time d_l^i, which corresponds to the deadline of session i; then step 2b identifies whether, for all d in the range $d_l^{i-1} < d \leq d_l^i$, the minimum $\min_{t>d} F_l(t) - 1(t-d)(\rho_0^m \cdot t)$ is obtained at time $t = d_l^i$. If so, then step 2c is executed. At that step, the algorithm identifies a potential point at which the slope changes to ρ_0^m, and a second point at which the slope changes to r_l^i. Otherwise, at step 2b only one potential point is identified, namely the one at which the slope changes to ρ_0^m.

input: $\{w_l^i, d_l^i, r_l^i\}_{0\leq i\leq I_l}$, (ρ_0^m, σ_0^m)
output: $\left\{y_l^j, x_l^j, a_l^j\right\}_{0\leq j\leq J_l}$
1. $k \leftarrow 0$, $\bar{x}_l^0 \leftarrow 0$, $\bar{a}_l^0 \leftarrow r_l^0$, $\bar{y}^0 \leftarrow 0$
2. for $i = 1$ to I_l:
   (a) $z^i \leftarrow \begin{cases} \min_{k>i} w_l^k - \rho_0^m\left(d_l^k - d_l^i\right), & i < I_l \\ w_l^i, & i = I_l \end{cases}$
   (b) if $z^i \leq w_l^i$
   i. $x \leftarrow \frac{z^i - w_l^{i-1} - \rho_0^m d_l^i + r_l^{i-1} d_l^{i-1}}{r_l^{i-1} - \rho_0^m}$
   ii. if $d_l^{i-1} < x < d_l^i$ then
   A. $k \leftarrow k+1$
   B. $\bar{x}_l^k \leftarrow x$, $\bar{a}_l^k \leftarrow \rho_0^m$, $\bar{y}_l^k \leftarrow y_l^{k-1} + \bar{a}_l^{k-1}\left(\bar{x}_l^k - \bar{x}_l^{k-1}\right)$
   (c) else if $z^i > w_l^i$
   i. $x \leftarrow \frac{w_l^i - w_l^{i-1} - \rho_0^m d_l^i + r^{i-1} d_l^{i-1}}{r_l^{i-1} - \rho_0^m}$
   ii. if $d_l^{i-1} < x < d_l^i$ then
   A. $k \leftarrow k+1$
   B. $\bar{x}_l^k \leftarrow x$, $\bar{a}_l^k \leftarrow \rho_0^m$, $\bar{y}_l^k \leftarrow y_l^{k-1} + \bar{a}_l^{k-1}\left(\bar{x}_l^k - \bar{x}_l^{k-1}\right)$,
   iii. $k \leftarrow k+1$
   iv. $\bar{x}_l^k \leftarrow d_l^i$, $\bar{a}_l^k \leftarrow r_l^i$, $\bar{y}_l^k \leftarrow y_l^{k-1} + \bar{a}_l^{k-1}\left(\bar{x}_l^k - \bar{x}_l^{k-1}\right)$
3. $k \leftarrow k+1$
4. $\bar{x}_l^k \leftarrow d_l^k$, $\bar{a}_l^k \leftarrow r_l^k$, $\bar{y}_l^k \leftarrow w_l^I$
5. $K \leftarrow k$.
6. if $\bar{y}_l^K \leq \sigma_0^m$
   (a) $x_l^0 \leftarrow \bar{x}_l^K + \frac{\sigma_0^m - \bar{y}_l^K}{\bar{a}_l^J}$, $a_l^0 \leftarrow \bar{a}_l^K$, $y_l^0 \leftarrow 0$
   (b) $J_l \leftarrow 0$
7. else
   (a) let b such that $\bar{y}_l^{b-1} \leq \sigma_0^m$ and $\bar{y}_l^b > \sigma_0^m$.
   (b) $j \leftarrow 0$, $x_l^0 \leftarrow \bar{x}_l^b - \frac{y_l^b - \sigma_0^m}{a_l^{b-1}}$, $a_l^0 \leftarrow \bar{a}_l^{b-1}$, $y_l^0 \leftarrow 0$.
   (c) for $k \leftarrow b$ to K:
   i. $j \leftarrow j+1$, $x_l^j \leftarrow \bar{x}_l^k$, $a_l^j \leftarrow \bar{a}_l^k$, $y_l^j \leftarrow \bar{y}_l^k - \sigma_0^m$
   (d) $J_l \leftarrow j$

Fig. 12. Algorithm Calculate-Residual-Resource

Proposition 8: (a) Let $\mathcal{I}_l$ be a schedulable set of sessions at link l, and let $\{w_l^i, d_l^i, r_l^i\}_{0\leq i\leq I_l}$ be a set of parameters defining the link work availability function. Also, let (ρ_0^m, σ_0^m) characterize a new session m such that the stability condition $\sum_{i\in\mathcal{I}} \rho_l^i + \rho_l^m < R_l$ is satisfied. Then, the link residual resource function $S_l(d)$, is fully specified by the set $\left\{y_l^j, x_l^j, a_l^j\right\}_{0\leq j\leq J_l}$, obtained by Algorithm Calculate-Residual-Resource.

(b) The algorithm's complexity is $O\left(I_l^2\right)$.

Proof: Algorithm Calculate-Residual-Resource iterates over the delays $\{d_l^i\}_{i\in\mathcal{I}_l}$ and identifies the corresponding points at which the slope of $S_l\left(d\right)$ changes. The algorithm implements the following scheme.

For each $i \in \mathcal{I}_l$, let

$$e = \arg\min_{k>i} F_l\left(d_l^k\right) - \rho_0^m\left(d_l^k - d_l^i\right).$$

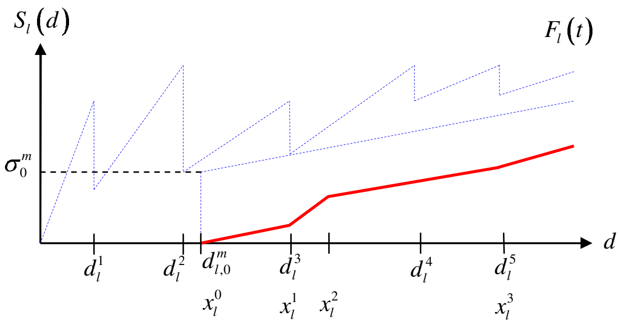
If $F_l\left(d_l^e\right) - \rho_0^m\left(d_l^e - d_l^i\right) \leq F_l\left(d_l^i\right)$ then find x such that

$$F_l\left(d_l^{i-1}\right) + r_l^{i-1}\left(x - d_l^{i-1}\right) = F_l\left(d_l^e\right) - \rho_0^m\left(d_l^e - x\right);$$

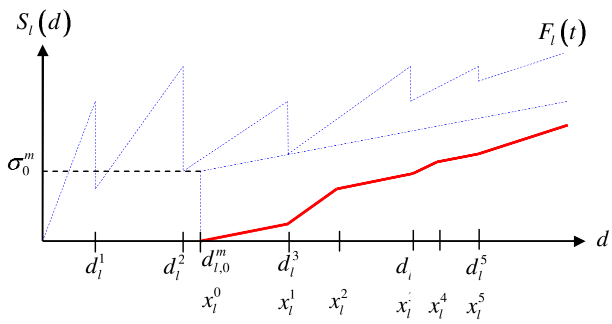
if $d_l^{i-1} < x < d_l^i$ then the slope of $S_l\left(d\right)$ changes at x (see, for example, point x_l^2 in Figure 13(a)). Otherwise ($F_l\left(d_l^e\right) - \rho_0^m\left(d_l^e - d_l^i\right) > F_l\left(d_l^i\right)$), find x such that

$$F_l\left(d_l^{i-1}\right) + r_l^{i-1}\left(x - d_l^{i-1}\right) = F_l\left(d_l^i\right) - \rho_0^m\left(d_l^i - x\right);$$

if $d_l^{i-1} < x < d_l^i$ then the slope of $S_l\left(d\right)$ changes at x (see, for example, point x_l^2 in Figure 13(b)). Furthermore, the slope of $S_l\left(d\right)$ changes again at d_l^i (see, for example, point x_l^3 in Figure 13(b))



(a)



(b)

Fig. 13. Typical instances of the residual resource function

One can see that the above scheme identifies all the points at which the slope of $S_l\left(d\right)$ changes as well as the function and the slope values at these points. Accordingly, the corresponding set $\left\{y_l^j, x_l^j, a_l^j\right\}_{0\leq j\leq J_l}$, identified by the algorithm, fully specifies $S_l\left(d\right)$.

For each i, $1 \leq i \leq I_l$, the algorithm consists $O\left(I_l\right)$ operations. Thus, the algorithm's complexity is $O\left(I_l^2\right)$. ■

One can see that the way by which the path selection and deadline allocation affect the available resources at the network links depends on the the value of the residual resource function $S_l\left(d_l\right)$. Therefore, a better measure for balancing the loads over the network might be one that accounts for the residual resources, rather than the residual rate. Accordingly, we aim at balancing the load by seeking a feasible path for which the residual resourced of its bottleneck link is maximal. That is, we seek a path that solve the following problem.

Maximum Residual Resource(MRR): *Given are a network $G\left(V,E\right)$, with a service rate R_l, a propagation delay δ_l, and a work availability function $F_l\left(t\right)$ for each $l \in E$. Also, given is a session m with source s^m,* *destination t^m, upper rate ρ_0^m, burst σ_0^m and an end-to-end delay requirement D^m. Find a feasible path-deadline assignment $(\mathbf{p}^m, \underline{d}^m)$,* i.e., *a path between s^m and t^m and a deadline allocation $(d_l^m)_{l\in\mathbf{p}^m}$, such that:*

1. $\sum_{l\in\mathbf{p}^m} (d_l^m + \delta_l) \leq D^m$,

2. for all $l \in \mathbf{p}^m$:

$$F_l(t) - 1(t - d_l^m)(\sigma_0^m + \rho_0^m(t - d_l^m)) > 0 \ \forall t > 0.$$

If there are several such paths, select the path-deadline assignment that solves the following: $\max_{\mathbf{p},(d_l)_{l\in\mathbf{p}}} \min_{l\in\mathbf{p}} (S_l(d_l))$.

Algorithm MRR, specified in Figure 14, identifies such a path. First the algorithm calculates for each link $l \in E$ the residual-resource function through algorithm Calculate-Residual-Resource. Then, it performs a binary search to find the maximal minimal allowed residual-resource value for which a feasible path exists. For each minimal allowed value, the algorithm calculates the minimum available deadline d_l^m at each link l (through the residual-resource function). Then, it finds the shortest path with respect to $\{d_l^m + \delta_l\}$. If the identified path is feasible, then the minimal residual-rate value can be increased, otherwise it should be decreased.

1. for all $l \in E$, call Calculate-Residual-Resource
2. for all $l \in E$, let the minimum delay $d_{l,0}^m \leftarrow x_l^0$
3. find the shortest path $\tilde{\mathbf{p}}$ between s^m and t^m w.r.t. $\left\{d_{l,0}^m + \delta_l\right\}$ (through Dijkstra's shortest path algorithm)
4. if $\sum_{l\in\tilde{\mathbf{p}}} \left(d_{l,0}^m + \delta_l\right) < D$ then
   (a) for all $l \in E$
   i. if $D > x_l^{K_l}$ then $R_l(D) \leftarrow y_l^{K_l} + a_l^{K_l}\left(D - x_l^{K_l}\right)$
   ii. else
   A. let b_l such that $x_l^{b_l-1} \leq D < x_l^{b_l}$
   B. $R_l(D) \leftarrow y_l^{b_l-1} + a_l^{b_l-1}\left(D - x_l^{b_l-1}\right)$
   (b) $L \leftarrow 0$ $H \leftarrow \max_l R_l(D)$, $\lambda \leftarrow H$
   (c) while $\left|\lambda - \frac{L+H}{2}\right| > 1$
   i. $\lambda \leftarrow \frac{L+H}{2}$
   ii. for all $l \in E$
   A. if $\lambda > y_l^{K_l}$ then $d_l^m \leftarrow x_l^{K_l} + \frac{\lambda - y_l^{K_l}}{a_l^{K_l}}$
   B. else let b_l such that $y_l^{b_l-1} \leq \lambda < y_l^{b_l}$,
   $d_l^m \leftarrow x_l^{b_l-1} + \frac{\lambda - y_l^{b_l-1}}{a_l^{b_l-1}}$
   iii. find the shortest path $\tilde{\mathbf{p}}$ between s^m and t^m w.r.t. $\{d_l^m + \delta_l\}$ (through Dijkstra's shortest path algorithm)
   iv. if $\sum_{l\in\tilde{\mathbf{p}}} (d_l^m + \delta_l) \leq D$ then $L \leftarrow \lambda$ else $H \leftarrow \lambda$
   (d) return $\tilde{\mathbf{p}}$, $\{d_l^m\}_{l\in\tilde{\mathbf{p}}}$
5. else there is no feasible path, end

Fig. 14. Algorithm MRR

Proposition 9: Algorithm MRR correctly solves the problem. The algorithm's complexity is

$$O\left(M \cdot I_{\max}^2 + \log(R'_{\max} D)(N \log N + M)\right).$$

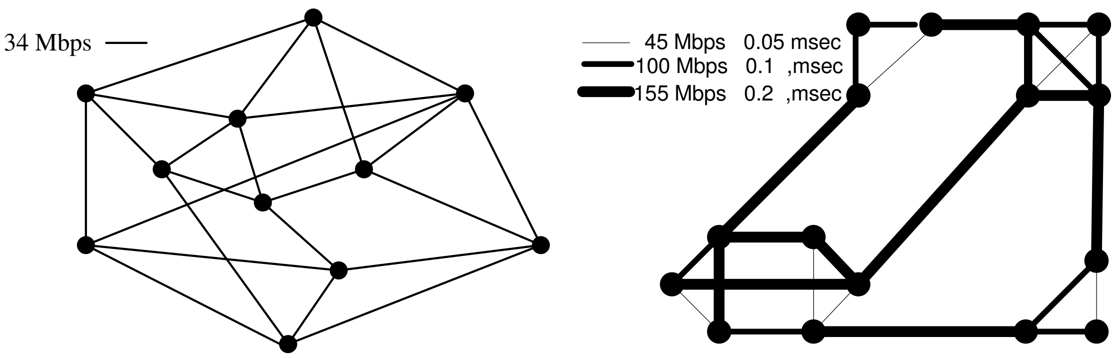
Proof: The first part of the proposition results from the fact that the residual-resource function is increasing.
The algorithm calculates the residual-resource for all links, which involves $O\left(MI_{\max}^2\right)$ operations. Then, the algorithm consists $O\left(\log\left(R'_{\max}D\right)\right)$ iterations; at each iteration Dijkstra's shortest path algorithm is executed. ■

VI. Simulation Results

We now demonstrate the efficiency of the proposed load-balancing routing schemes by way of simulations. Our figure of merit is session blocking probability, which is evaluated for various loads and network topologies.

Following [19], the characteristics of the sessions that need to be supported in the network are generated randomly and are intended to cover a wide range of traffic patterns. We take $\rho = 10^p Kb/s$, where p is uniformly distributed in $[1, 3]$. Hence, ρ covers the range $[10Kb/s, 1Mb/s]$. Next, we take $\sigma = r \cdot \rho Kb$, where r is uniformly distributed in $[0.8, 1.6]$. Session requests are generated according to a Poisson process with a parameter α. The session's source and destination nodes are uniformly chosen. A session is accepted if the considered routing scheme identifies a feasible route; otherwise, the session is rejected. An accepted session stays in the system for an exponentially distributed duration time with mean $\frac{1}{\beta}$. The ratio $\frac{\alpha}{\beta}$ characterizes the traffic load offered to the network, *i.e.*, the average number of flows that would exist at any time at a network with no capacity limitation. The blocking probability is evaluated as the total number of rejected sessions divided by the total number of generated sessions.

Each session has a delay requirement $D = 10^s \cdot 30ms$, where s is uniformly distributed in $[0, 1.52]$, thus D ranging in $[30ms, 1s]$. We conducted the simulations on the network topologies depicted in Figure 15, which were extensively considered in network performance studies.



(a)COST239 topology (b)Clustered topology

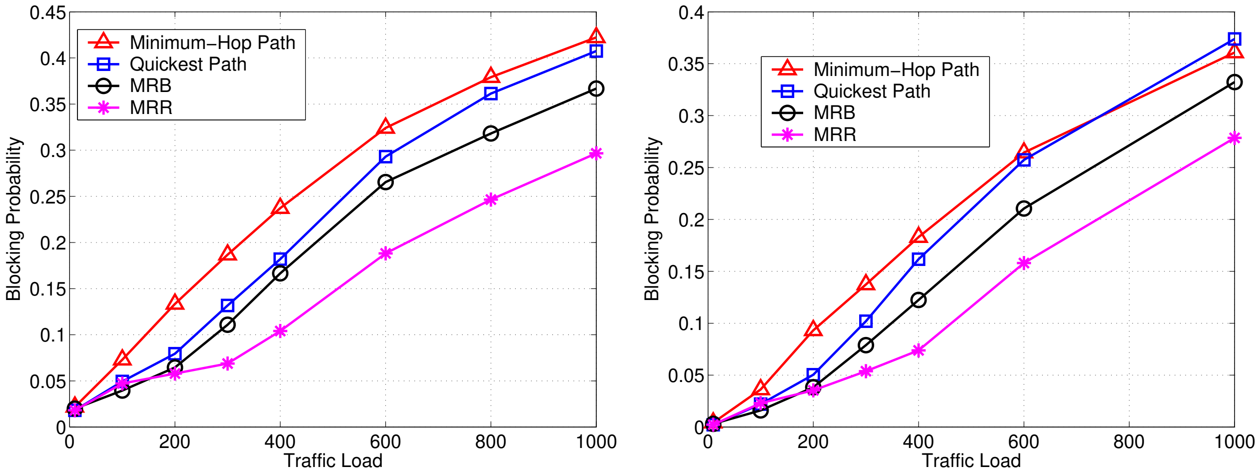
Fig. 15. Network topologies

We compared between the following routing algorithms:

- Our maximum residual bottleneck algorithm (MRB), which identifies a feasible path with the maximum residual rate of its bottleneck link.
- Our maximum residual resource algorithm (MRR), which identifies a feasible path with the maximum

residual resource of its bottleneck link.

- The quickest path algorithm, which identifies a feasible path with the minimum end-to-end delay bound.
- The minimum hop path algorithm, which identifies a feasible path with the minimum number of hops.



(a)COST239 topology (b)Clustered topology

Fig. 16. Session request blocking probability

We generated a total of $10,000$ sessions in each simulation run, *i.e.*, for each routing scheme and each traffic load. Each simulation run was repeated 5 times with different seeds. Fig. 16 presents the simulation results, in terms of the blocking probability as a function of the traffic load, in the two network topologies. One can see that our maximum residual resource (MRR) and maximum residual bottleneck (MRB) routing schemes outperform both the quickest path as well as the minimum hop path routing schemes. In particular, focusing on a load region for which the blocking percentage is 10% (*i.e.*, a significant yet still reasonably acceptable value), our maximum residual resource (MRR) algorithm admits about 60% more sessions than the quickest path routing scheme in the COST239 topology, and about 80% more sessions in the Clustered topology. Furthermore, Algorithm MRR admits over 100% more sessions than the minimum-hop path routing scheme, in both topologies.

VII. Conclusion

This study has considered QoS routing in networks that employ the rate-controlled EDF scheduling discipline.

First, we focused on the basic problem of identifying feasible paths. Here, we broadened the space of feasible solutions by allowing to reshape the traffic with different parameters at each hop. Accordingly, we established an optimal routing scheme that considered the joint problem of identifying a feasible path and optimizing the reshaping parameters along the path. However, the computational complexity of this scheme might be prohibitively large. Thus, we established an approximation scheme that is based on quantizing the reshaping delay, which is ϵ-optimal and at the same time computationally efficient. Our scheme is guaranteed to find a feasible solution (*i.e.*, path) whenever the standard EDF routing schemes do so. Moreover, we demonstrated that our scheme identifies feasible solutions also in cases in which the standard schemes fail.

We also consider the special case in which only the maximal burst size of the traffic can be reshaped along the path. For this case, we establish an optimal routing scheme as well as an ϵ-optimal solution of lower complexity.

Next, we considered the more complex problem, of optimizing the route choice in terms of balancing the loads and accommodating multiple connections. Here, we established and validated two routing schemes. The first scheme aims at balancing the load by identifying a feasible path (if one exists) with a maximum bottleneck residual rate. The second scheme jointly considers the problems of route selection and assignment of deadlines along the chosen path. Clearly, considering the problems jointly, instead of independently as in previous approaches, leads to a better utilization of the network resources. Simulation results demonstrated the advantages of our schemes, in particular the second one.

To sum up, this study has investigated two novel classes of EDF routing schemes: the first allows per-hop optimization of the reshaping parameters, while the second aims to optimize the global utilization of network resources. An important yet complex task, left for future research, is to *merge* the two classes, *i.e.*, establish an efficient EDF routing scheme that performs per-hop reshaping optimization *and* optimizes the utilization of the network.

REFERENCES

[1] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks - the single node case," *IEEE/ACM Transanctions on Networking*, vol. 1, 1993.

[2] Z. Wang and C. Crowcroft, "QoS routing for supporting resource reservation," *IEEE JSAC*, September 1996.

[3] R. Guerin and A. Orda, "QoS-based routing in networks with inaccurate information: Theory and algorithms," *IEEE/ACM Transanctions on Networking*, vol. 7, no. 3, pp. 350–364, June 1999.

[4] Q. Ma and P. Steenkiste, "Quality of service routing for traffic with performance guarantees," in *IWQOS'97*, May 1997, pp. 115–126.

[5] A. Orda, "Routing with end to end QoS guarantees in broadband networks," *IEEE/ACM Transanctions on Networking*, vol. 7, no. 3, pp. 365–374, June 1999.

[6] E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick, "A framework for QoS-based routing in the internet," RFC 2386, Internet Engineering Task Force, August 1998.

[7] H. Zhang and D. Ferrari, "Rate-controlled service disciplines," *Journal of High Speed Networks*, vol. 3, no. 4, pp. 389–412, 1994.

[8] L. Georgiadis, R. Guerin, and A. Parekh, "Optimal multiplexing on a single link: Delay and buffer requirements," *IEEE Trans. Information Theory*, vol. 43, no. 5, pp. 1518–1535, Septemer 1997.

[9] L. Georgiadis, R. Guerin, V. Peris, and K. N. Sivarajan, "Efficient network QoS provisioning based on per node traffic shaping," *IEEE Trans. Networking*, vol. 4, no. 4, pp. 482–501, August 1996.

[10] J. Liebeherr, D. Wrege, and D. Ferrari, "Exact admission control for networks with a bounded delay service," *IEEE/ACM Transanctions on Networking*, vol. 4, no. 6, pp. 885–901, December 1996.

[11] V. Firoiu, J. Kurose, and D. Towsley, "Efficient admission control of piecewise linear traffic envelopes at edf schedulers," *IEEE/ACM Transanctions on Networking*, vol. 6, no. 5, pp. 558–570, October 1998.

[12] V. Sivaraman, F. M. Chiussi, and M. Gerla, "Traffic shaping for end-to-end delay guarantees with edf scheduling," in *IWQoS 2000*, 2000.

[13] D. Ferrari and D. C. Verma, "A scheme for real-time channel establishment in wide-area networks," *IEEE Journal of Selected Areas in Communication*, vol. 8, pp. 368–379, April 1990.

[14] Chih-Che Chou and Kang G. Shin, "A distributed table-driven route selection scheme for establishing real-time video channels," in *International Conference on Distributed Computing Systems*, 1995, pp. 52–59.

[15] A.S. Ayad, M.T. El-Hadidi, and K.M.F. Elsayed, "Resource division policies for edf scheduling in atm networks," in *Proc. of the IEEE ISCC'01*, July 2001, number 3-5, pp. 249 – 254.

[16] K.M.F. Elsayed, A.S. Ayad, and M.T. El-Hadidi, "Evaluation of resource reservation policies for deterministic services in multi-service packet networks," in *PROCEEDINGS OF 18th International Teletraffic Congress*, September 2003, pp. 1081–1090.

[17] R. L. Cruz, "A calculus for network delay, part i: Network elements in isolation," *IEEE Trans. Information Theory*, vol. 37, no. 1, pp. 114–131, January 1991.

[18] Kang G. Shin amd Chih-Che Chou and Seok-Kyu Kweon, "Distributed route selection for establishing real-time channels," *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, no. 3, pp. 318 – 335, March 2000.

[19] V. Firoiu, J. Kurose, and D. Towsley, "Efficient admission control for EDF schdulers," CMPSCI Technical Report TR 96-46, Department of Computer Science, University of Massachusetts, MA, USA, April 1997. Available from ftp://ftp.cs.umass.edu/pub/techrept/techreport/1996/UM-CS-1996-046.ps.