Tuning the weights in WFQ schedulers for the maximization of carried best effort traffic

Eduardo Magaña, Daniel Morató, Pravin Varaiya
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, CA 94720
email: {emagana, dmorato, varaiya}@eecs.berkeley.edu

Abstract—
This paper shows a configuration scheme for networks with WFQ schedulers that maximizes the amount of best effort traffic carried. We focus on those cases where traffic flows would congest critical links and lower network performance. The proposal is based on a Linear Programming formulation. The solution provides the weights that will allow us to control the best effort flows and reach the optimal situation. We offer a formulation that reaches a trade-off between network utilization, fairness, and user satisfaction.

Keywords— **Traffic engineering, Traffic control, Network performance**

I. Introduction

Providing Quality of Service (QoS) requirements for certain flows in "best effort" IP networks is a topic of attention from researchers, enterprises and Internet Service Providers (ISPs).

Solutions based on DiffServ (Differentiated Services) [1] or IntServ (Integrated Services) [2] provide mechanisms to guarantee certain throughput and delay to the flows with QoS constraints in an individual Autonomous System [3]. They focus on flows that we call EF (Expedited Forwarding) traffic. The remaining available bandwidth is used by the so-called Best Effort (BE) traffic.

For the provision of this QoS, new schedulers are been implemented in network routers. Schedulers like WFQ (Weighted Fair Queuing [4]), PGPS (Packetized Generalized Processor Sharing [5]) and CBQ (Class Based Queuing [6]) can provide a minimum bandwidth for required flows. The configuration of the schedulers is straight forward from the requirements of these EF flows. However, these routers typically use this scheduling mechanism with BE traffic too. The default configuration gives the same weight to every flow or a weight based on the TOS bits in the IP header. There is a lack of an accepted solution for the configuration of weights for these flows without requirements, a solution that could be applied to the huge variability of services and traffic types found in data networks. Even for the flows from services that carry a large percentage of the network traffic, it is not easy to optimize their impact on the network.

In this paper we present a simple way solve this configuration. The goal is to optimize network use from the point of view of the service provider. This provider will try to maximize his revenue. As best effort flows, by definition, don't have any specific quality requirement, this approach will look for those flows that make the best use of network resources. But even for BE traffic, we include an objective of fairness among flows and we measure its impact on the maximum revenue. This fairness brings the user's point of view to the study and avoids the starvation of some flows.

Other proposals have focused mainly in routing algorithms with QoS, trying to find the best routes for EF traffic [7] [8]. The best routes will be those less congested, with less delay or those that would minimize blocking probability for future arriving flows. In this paper we assume that path selection for any kind of EF traffic is solved by a known method. Once the EF traffic is routed there is still a large amount of BE traffic using the residual available bandwidth. Typically, this available bandwidth has been managed by the routing protocol for best effort traffic [9]. We show that even using a shortest-path routing protocol, a big improvement can be achieved selecting the optimal bandwidth resources for each BE flow. This sharing becomes interesting when there is congestion in the network and so, not every packet could be carried. In this situation a bad selection of flows could congest some critical paths in the network and starve many other flows, moving the working point of the network to a far from optimal situation. We will focus into maximizing carried traffic for the worst cases of congestion, when the sharing policy becomes critical.

We assume that a flow-based multiplexing and scheduling discipline is available in each router. The packet scheduler will give priority to EF traffic. Among the EF flows we can share the bandwidth configuring the *weight* assigned to each flow. For the best effort traffic we will use precomputed weights that try to select the optimal flows. We will try to set up these weights for the BE flows in such a way that the carried traffic will be as high as possible. As far as we know the literature does not address the problem of providing optimal WFQ weights for the BE traffic.

Using a WFQ scheduler for BE traffic means that the nodes will provide a minimum bandwidth for the BE flows. This could look like contradictory with the definition of "best effort" traffic, but we should remember that the scheduling discipline is work conserving and so the unused bandwidth of a flow will never be wasted while there are queued packets.

To calculate these BE weights, we will use a Linear Programming (LP) approach, trying to maximize the load in the network. This approach has been successfully used in

This research was supported by DARPA Contract N66001-00-C-8062

similar flow maximization problems [10] [11] [12].

The rest of the paper is organized as follows: in section II we present the scenario. Section III presents the formulation of the proposal. Section IV is devoted to an in-depth analysis of the results and finally section V presents the conclusions that can be drawn from this study.

II. Scenario

Our network scenario is any topology of nodes (routers) interconnected by links with different bandwidth. Every node is supposed to be in the same administrative domain so the global information of the topology is known and the configuration of the flows in the nodes is not an issue. In this network there can be EF flows and BE traffic. However, once the paths for the EF flows are known, a minimum bandwidth for these flows is guaranteed and unused bandwidth is left available to other classes. This guaranteed bandwidth is the minimum provided by the WFQ scheduler in the case of congestion. For the BE traffic we study all the traffic from node A to node B as only one total BE flow $A \rightarrow B$. The routes for BE traffic will be assumed as static during the calculations and given by any routing algorithm based on shortest paths [13]. An important difference with other optimization works in the literature should be highlighted: the traffic matrix is not an input parameter. The optimization problem finds the best arrangement of BE flows that will maximize the carried traffic. The solution provides the bandwidth that should be enforced for the best effort flows when the sources are greedy.

The nodes in the topology could be traffic sources and/or sinks. We add the category of *transient nodes*. A transient node is neither source nor destination, and is used to model the routers not attached to any network with hosts.

We are interested in networks built with generalized processor sharing schedulers (PGPS or WFQ, *packetized* versions of GPS). From [5], for each backlogged session i throughout the time interval $(\tau, t]$, GPS is defined as the scheduling such that:

$$\frac{S_i(\tau,t)}{S_j(\tau,t)} \geq \frac{\phi_i}{\phi_j}, j = 1, 2, \ldots N \tag{1}$$

where $S_i(\tau,t)$ is the amount of session i traffic served in that interval and ϕ_i are the weights. It provides a guaranteed rate for session i of:

$$g_i = \frac{\phi_i}{\sum_j \phi_j} r_i \tag{2}$$

where r_i is the session i average rate. Additionally, it provides worst-case network queuing delay guarantees when the sources are constrained by leaky buckets.

We work with a worst case total congestion in the network. In this situation each flow receives the minimum bandwidth provided by the configured weights. The proposed approach will calculate the weights for the BE traffic that maximizes carried traffic. At the same time, as a trade-off, we will try to provide some *fairness*. This fairness will avoid the starvation of some flows.

III. Methodology

In this section we specify the constraints and objective function for the linear program that will provide the weights for the best effort flows.

A Linear Program in standard form follows equations 3a-3c, where x is a column vector with the unknown variables to be solved, $\mathbf{A}$ is a matrix of coefficients and b and c are column vectors with more coefficients. The bounds in equation 3b can be generalized. The objective function 3c can be turned into a maximization and the problem can be solved using standard techniques like the traditional Simplex method.

$$\mathbf{A}x = b \tag{3a}$$

$$x \geq 0 \tag{3b}$$

$$\text{minimize } cx \tag{3c}$$

For the formulation of this particular problem let N be the set of nodes in the network and L the set of links ($L \subseteq N \times N$ and $\|L\|$ is the number of elements in L). Each node could have one link (end node or host) or several links with other nodes (router). Each link is a pair $z = (x, y) \in L$ where $x, y \in N$. Let $b_{s,d}$, $(s, d \in N)$ be the amount of traffic carried from node s to node d (not necessarily adjacent ones). We call this flow $s \rightarrow d$ and $Path_{s \rightarrow d}$ is the set of links in the path from node s to node d (eq. 4). This path is calculated by a routing protocol and we keep it fixed for our calculations.

$$Path_{s \rightarrow d} = \{(s, n_0), (n_0, n_1) ... (n_k, d)\} \tag{4}$$

If N_t is the set of transient nodes, every $b_{s,d}$ flow with any of the end nodes (s and/or d) in the set of transient nodes must be 0. We express this with the set of constraints:

$$b_{s,d} = 0 \quad \forall s, d \in N / s \in N_t \quad \text{and/or} \quad d \in N_t \tag{5}$$

One of the main constraints is based on the fact that the amount of traffic in a link $z = (n, m) \in L$ must be limited by the available bandwidth in that link. We denominate BW_z the available bandwidth for best effort traffic in a link z once the configuration for the EF flows has been done. For each link z there is a subset of flows $F \subseteq N \times N$, such as z belongs to the path of every flow in F ($\forall (s, d) \in F$, $z \in Path_{s \rightarrow d}$). That means that all those flows $s \rightarrow d$ use link z and consume bandwidth from BW_z. We express this constraint in the form of the set of equations:

$$\sum_{s,d \in N / z=(n,m) \in Path_{s \rightarrow d}} b_{s,d} \leq BW_z, \quad b_{s,d} \geq 0 \tag{6}$$

For the last set of constraints we define an auxiliary variable K. This variable is the minimum amount of bandwidth assigned to each BE flow (eq. 7). We will get a solution with a trade-off between user goals and administrator goals.

$$\forall s, d \in N - N_t \quad b_{s,d} \geq K \geq 0 \tag{7}$$

With these constraints we analyze three problems based on three different objective functions for the linear program. Namely:

(a) Maximum traffic: The objective is to maximize the amount of traffic carried by the network.

$$Objective = \max\{\sum_{s,d \in N/s \neq d} b_{s,d}\} \quad (8)$$

(b) Providing minimum bandwidth: We solve two LPs, both with the same set of constraints. The first one uses the following objective function:

$$Objective = \max\{K\} \quad (9)$$

The solution is a minimum and equal amount of bandwidth for every possible flow in the network. For the second LP we subtract the bandwidth used by the flows calculated in the first step from BW_z. With the network comprising the remaining link bandwidth we formulate the goal of maximum network use. Using the same procedure as in method (a) we formulate (10) as the objective function for the new LP with this network and the same set of constraints. The variables $\hat{b}_{s,d}$ in this equation have the same meaning as the $b_{s,d}$ in the previous formulation but are specific for this particular situation where the minimum bandwidth has already been subtracted from the topology. This second LP follows the same approach as method (a) but from a different starting point.

$$Objective = \max\{\sum_{s,d \in N/s \neq d} \hat{b}_{s,d}\} \quad (10)$$

In this method the total configured bandwidth per flow is $b_{s,d} = K + \hat{b}_{s,d}$ $(s, d \notin N_t)$.

(c) Trade-off: In this approach we study the effect of an objective function that combines cases (a) and (b). We solve an LP with equation 11.

$$Objective = \max\{\sum_{s,d \in N - N_t/s \neq d} \hat{b}_{s,d} + \alpha K\} \quad (11)$$

where α is an independent coefficient that controls the effect of variable K in the problem. In section IV-B we investigate the effects from this α coefficient.

We denote with a superscript the solution of the linear programs in each of the three methods, $b_{s,d}^{(\beta)}$, $K^{(\beta)}$, $\beta \in \{a, b, c\}$. Finally we define $T_{BE}^{(\beta)}$ as the total amount of BE traffic carried by the network with the solution from method (β):

$$T_{BE}^{(\beta)} = \sum_{s,d \in N/s \neq d} b_{s,d}^{(\beta)} \quad (12)$$

These $b_{s,d}^{(\beta)}$ coefficients turn into the ϕ_i weights in the packet schedulers (eq. 1) for the flows from each node s to each node d. They have to be configured in each node in $Path_{s \to d}$.

IV. Analysis

In this section we analyze the solutions obtained from the linear programs formulated in section III.

A. Basic scenarios

In order to explain the observed effects we take three basic topologies. These test topologies can be seen in Fig. 1, where the transient nodes are filled with a gray pattern. All the links have capacity 1.

We use several parameters for the evaluation of the result from each LP proposed. The total BE traffic carried is our primary goal. Another parameter we present is the minimum bandwidth reserved for each flow. We can get this value from variable K in methods (b) and (c). However, in method (a) we could have $K = 0$ and at the same time get a minimum bandwidth greater than 0. The reason for this result is that the objective function for method (a) doesn't use variable K. The solving method for the LP could choose $K = 0$ because that way the constraints are true and changing the value of K doesn't change the result of the objective function. So for method (a) we get the minimum reserved bandwidth looking at all the $b_{s,d}^{(a)}$. We also show a measurement of disparity D among flows, calculated with equation 13. $B^{(\beta)}$ is the average bandwidth for the BE flows (average of the $b_{s,d}^{(\beta)}$) and so $D^{(\beta)}$ is the average difference from the $b_{s,d}^{(\beta)}$ to $B^{(\beta)}$ (squared like a variance estimator). In the absence of any preference among BE flows, we use D to measure the *fairness* in the result. This disparity is not an absolute measurement in the sense that we can not use it to compare different topologies, but it's an interesting figure when we use different methods to solve the configuration for the same topology. Among different topologies the variations in connectivity and link bandwidths make this parameter useless. Finally we show the percentage of flows with an assignment greater than 0. The minimum bandwidth, the disparity D and the percentage of non-null flows are not an objective of the network operator but they are quality measurements from a user's point of view. We use $\max\{T_{BE}\}$ as the goal of the network operator.

$$B^{(\beta)} = \frac{T_{BE}^{(\beta)}}{\|\{(s,d)/s, d \in N - N_t, s \neq d\}\|}$$
$$D^{(\beta)} = \frac{\sum_{s,d \in N - N_t/s \neq d} (b_{s,d} - B^{(\beta)})^2}{\|\{(s,d)/s, d \in N - N_t, s \neq d\}\|} \quad (13)$$

The three basic scenarios we study are a full connected network (table I), a ring (table II) and a tree with transient nodes (table III).

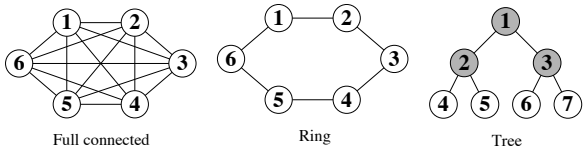

Fig. 1. Basic simulated scenarios

We can show the following insights from the analysis of these scenarios:

- Best utilization: Methods (a) (Max.Traff) and (c) (Trade-off with $\alpha = 1$) always get the best result in terms of maximum network utilization ($T_{BE}^{(\beta)}$).
- Minimum bandwidth: Method (a) sacrifices K. Some flows could get a 0 bandwidth assignment. Method (b) (Min.BW) always provides a minimum bandwidth to all the flows and the highest that is allowed. Method (c) works as (a) or as (b) in different topologies.
- Trade-off: Method (c) tries to provide a minimum bandwidth at the same time that maximizes network use. In section IV-B we will describe the effect of the α parameter. This parameter controls the effect each part of the objective function has on the solution. We will show that method (c) behaves as method (a) or (b) as α goes to 0 or to ∞.
- Minimum disparity: Method (b) gets the smallest disparity D in every case. The flows receive bandwidth assignments closer to each other and so the sharing is fairer. We find that the maximization of variable K in an independent step has not only an effect on minimum bandwidth but also reduces this variability. By configuring this minimum we get a closer bandwidth distribution.
- Shortest-path flows: If we look at the best $b_{s,d}^{(a)}$ flows we see that they are those with the shortest paths. In topologies without transient nodes we can fill all the links using just one-hop flows. For example, in the ring case described, all the $b_{s,d}^{(a)} > 0$ use one-hop routes. In topologies with transient nodes, like the tree case, we may need longer flows.
- Effect of K: Method (b) first tries to provide a minimum bandwidth for every possible flow in the network. It forces all the flows to be greater than zero, even if they use long, non one-hop paths. Providing a minimum bandwidth for every flow is done at the expense of link bandwidth that could have been used for other flows. In any non trivial topology we may find pairs of *(source_node, destination_node)* whose path is longer than one hop. In these cases we are reserving the provided bandwidth for that flow in more than one link. If not all the nodes in the path between *source_node* and *destination_node* are transient, that means that there are other flows with shorter paths that could use that bandwidth with a smaller expense of total bandwidth in the network. That way, with the same amount of link bandwidth expense we would get a larger total carried traffic $T_{BE}^{(\beta)}$. That's the reason method (b) reaches a lower $T_{BE}^{(\beta)}$. This can be easily seen with the help of Fig. 2. In this simple topology there aren't transient nodes. The possible flow pairs are $b_{1,2}$, $b_{2,1}$, $b_{2,3}$, $b_{3,2}$, $b_{1,3}$ and $b_{3,1}$. Providing a minimum and equal bandwidth K to all of them means configuring K even for flows $1 \rightarrow 3$ and $3 \rightarrow 1$. Flow $1 \rightarrow 3$ uses K in the links $(1, 2)$ and $(2, 3)$. That means that while in T_{BE} we count only K we expend $2K$. If the minimum bandwidth constraint doesn't apply then the flows $1 \rightarrow 2$ and $2 \rightarrow 3$ can be configured with a $2K$ bandwidth and with the same expense in total bandwidth the carried traffic is increased by K. The same procedure can be applied to the $3 \rightarrow 1$ flow. This is the reason in some topologies minimum bandwidth assignments don't take to the maximum T_{BE}.

TABLE I
RESULTS FOR FULL CONNECTED TOPOLOGY ($\alpha = 1$)

Method	T_{BE}	min BW	$D^{(\beta)}$	% $b_{s,d} > 0$
Max.Traf.	30	1	0	100
Min.BW	30	1	0	100
Trade-off	30	1	0	100

TABLE II
RESULTS FOR RING TOPOLOGY ($\alpha = 1$)

Method	T_{BE}	min BW	D	% $b_{s,d} > 0$
Max.Traf.	12	0	0.248	40
Min.BW	8	0.1667	0.041	100
Trade-off	12	0	0.248	40

TABLE III
RESULTS FOR TREE TOPOLOGY ($\alpha = 1$)

Method	T_{BE}	min BW	D	% $b_{s,d} > 0$
Max.Traf.	4	0	0.242	33
Min.BW	4	0.25	0.015	100
Trade-off	4	0.25	0.015	100

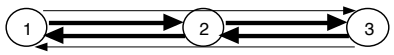

Fig. 2. Example of the effect of K

B. Complex scenario

In order to show the behavior of the three LP proposals, in this section we use a more general network topology. Fig. 3 presents a network with different link bandwidths, several bottlenecks and transient nodes. The number associated with each link is the available bandwidth. Again, transient nodes are filled with a gray pattern.

In table IV we show results similar to those we offered for the basic scenarios. Methods (a) and (c) (with $\alpha = 1$) get the maximum carried traffic (T_{BE}). Method (b) gets a lower utilization but it is the only one that provides a minimum bandwidth for every possible BE flow in the network. The smallest disparity among flows is reached using method (b).

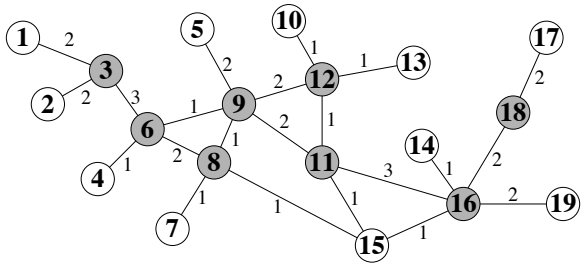

Fig. 3. Complex scenario

TABLE IV
RESULTS FOR COMPLEX TOPOLOGY

Method	T_{BE}	min BW	$D^{(\beta)}$	% $b_{s,d} > 0$
Max.Traf.	18	0	0.138	16
Min.BW	16.8	0.0625	0.051	100
Trade-off	18	0	0.156	16

Method (c), while providing the same T_{BE} and percentage of non null flows, gets the worst disparity.

We study now the effect of coefficient α that multiplies variable K in method (c). In Fig. 4, 5 and 6 we plot $T_{BE}^{(c)}$, the minimum bandwidth configured for each flow and the disparity $D^{(c)}$ as a function of the weight α.

When α is low, K is not as important as T_{BE} in the objective function. It's better (in terms of the objective function) to maximize the carried traffic than providing a minimum bandwidth for every flow. That's the reason the LP may find a better solution that sacrifices K in order to configure shorter flows with higher assignments. These shorter flows will carry more end-to-end traffic than longer ones with the same network use. Below certain interval of α we mainly find the same behavior as with method (a).

When α is high, the minimum bandwidth imposed by K is more important than carrying more flows. The LP will try to get the best minimum assignment and then it will continue maximizing T_{BE}. Even with high α, getting higher $b_{s,d}$ we can still improve the result of the objective. This way method (c) behaves as method (b) when working above a certain interval of α.

The steep transition from one kind of solution to the other is shown in the right side of Fig. 4, 5 and 6. As the network topology becomes larger and with many more different paths and bandwidths it would be easier to find a better solution focusing on the maximization of the T_{BE} part. We would need a higher α coefficient in order to prioritize K in the objective function. This would move the transition point in the figures to the right.

From Fig. 4 and 5 we can see that there are intervals of α where we get the same T_{BE} and the same minimum bandwidth (for this topology for example when $\alpha \in [0, 15]$). But sometimes those intervals show a changing disparity D (Fig. 6). If we look at the set of $b_{s,d}^{(c)}$ that result from the LP with different values of α we find that the solutions are different, all of them provide the same T_{BE} and the same minimum BW but the sharing of bandwidth among flows changes. All of these solutions reach the same value in the objective function so from the point of view of the linear program they are equivalent. Depending on the way the LP is solved and the initial step is chosen in the algorithm, we get different but equivalent solutions. This different sharing has been easily detected thanks to the parameter D that we defined.

V. CONCLUSIONS

We have shown that carried traffic (and so the revenue) can be improved choosing the optimal bandwidth for the best effort flows. The bandwidth in this optimization has been calculated using a linear program. The results trans-

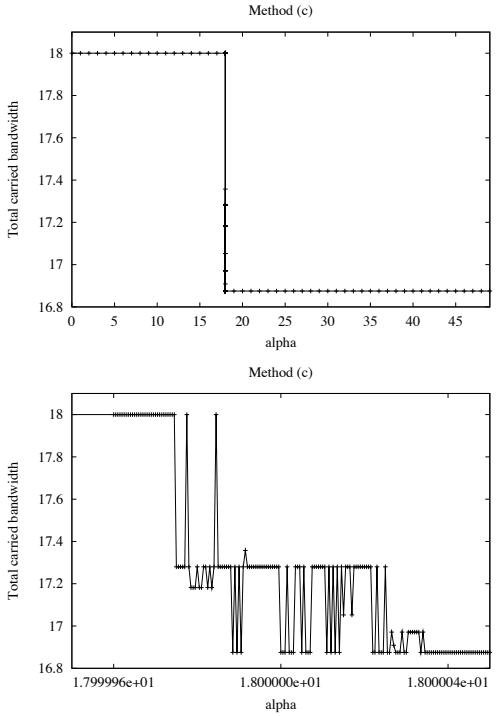

Fig. 4. T_{BE} versus the weight α

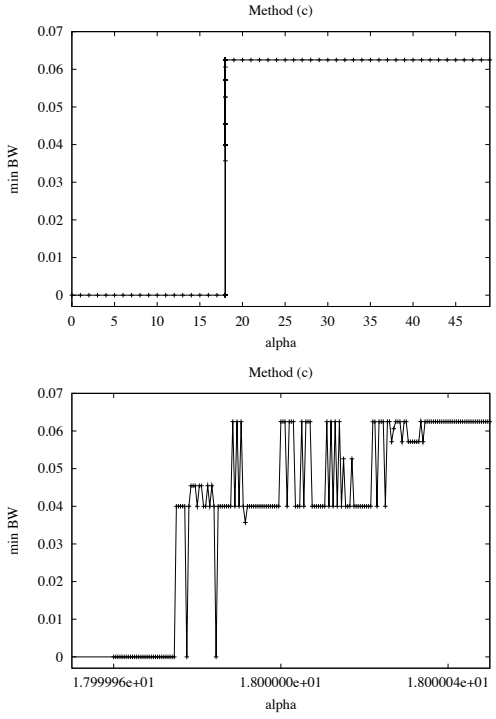

Fig. 5. Minimum BW versus the weight α

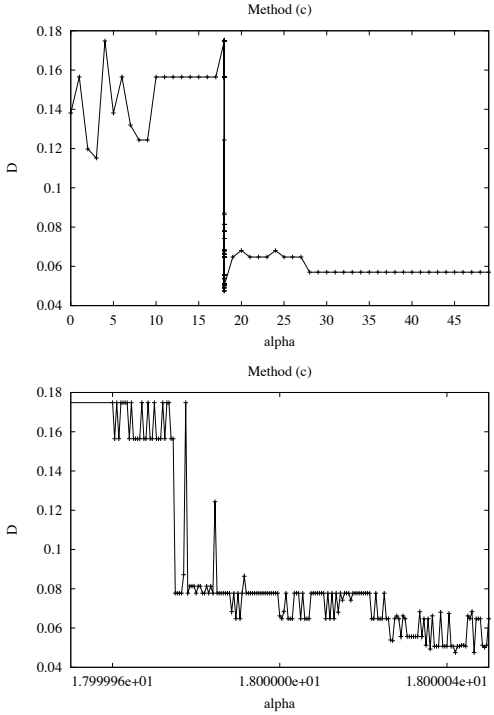

Fig. 6. D versus the weight α

late directly into the configuration of flow schedulers in the network routers. We have solved the problem of choosing the best WFQ weights for flows without specific QoS requirements. A requirement on optimal minimum bandwidth per flow can be added and it improves user satisfaction and fairness without increasing complexity in the formulation.

REFERENCES

[1] J. Liebeherr and N. Christin. Rate allocation and buffer management for Differentiated Services. *Computer Networks, Special Issue on the New Internet Architecture*, August 2002.

[2] P. White. RSVP and integrated services in the Internet: A tutorial. *IEEE Communications Magazine*, pages 100–106, May 1997.

[3] X. Xiao and L. M. Ni. Internet QoS: A big picture. *IEEE Network*, 13(2):8–18, Mar 1999.

[4] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. *Journal of Internetworking Research and Experience*, pages 3–26, October 1990.

[5] A. Parekh and R. Gallager. A Generalized Processor Sharing approach to flow control in Integrated Services networks: The single node-case. *IEEE/ACM Transactions on Networking*, 1(3), June 1993.

[6] S. Floyd and V. Jacobson. Link-sharing and resource management models for packet networks. *IEEE/ACM Transactions on Networking*, 3(4):365–386, August 1995.

[7] S. Chen and K. Nahrstedt. An overview of Quality-of-Service routing for the Next Generation high-speed networks: Problems and solutions. *IEEE Network, Special Issue on Transmission and Distribution of Digital Video*, Nov./Dec. 1998.

[8] A. Orda. Routing with end to end QoS guarantees in broadband networks. In *IEEE Infocom*, pages 27–34, 1998.

[9] Q. Ma, P. Steenkiste, and H. Zang. Routing high-bandwidth traffic in max-min fair share networks. In *ACM SIGCOMM96*, pages 226–217, August 1996.

[10] V. Chvatal. *Linear Programming*. Freeman, NY, 1983.

[11] C. Qiao and D. Xu. Distributed partial information management (DPIM) schemes for survivable networks - part I. In *IEEE Infocom*, 2002.

[12] R. Ramaswami and K.N. Sivarajan. Routing and wavelength assignment in all-optical networks. *IEEE Transactions on Networking*, pages 489–500, October 1995.

[13] A. V. Goldberg. Scaling algorithms for the shortest paths problem. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 222–231, January 1993.