Router scheduling configuration based on the maximization of benefit and carried Best Effort traffic

Eduardo Magaña (eduardo.magana@unavarra.es) and Daniel Morató (daniel.morato@unavarra.es)*

Department of Automatics and Computer Sciences

Public University of Navarra, 31006 Pamplona, Spain

Pravin Varaiya (varaiya@eecs.berkeley.edu)

Department of Electrical Engineering and Computer Sciences

University of California, Berkeley, CA 94720

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

Abstract. This paper shows a configuration scheme for networks with WFQ schedulers. It guarantees maximum revenue for the service provider in the worst case of network congestion. We focus on best effort traffic and select those flows that maximize the benefit while keeping the network utilization high. We show that optimum network configuration is feasible based only on knowledge of the topology. Its dependence on the pricing scheme can be reduced and even eliminated. We offer a formulation that reaches a tradeoff between network utilization, fairness, and user satisfaction.

Keywords: traffic engineering, traffic control, network performance

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) (Liebeherr and Christin, 2002) or IntServ (Integrated Services) (White, 1997) provide mechanisms to guarantee certain throughput and delay to the flows with QoS constraints in an individual Autonomous System (Xiao and Ni, 1999). They focus on two classes of traffic: flows with quality of service requirements (that we will call EF or Expedited Forwarding) and Best Effort (BE) traffic.

For the provision of this QoS, new schedulers have been implemented in network routers. Schedulers like Weighted Fair Queuing (WFQ) (Demers et al., 1990), Packetized Generalized Processor Sharing (PGPS) (Parekh and Gallager, 1993) and Class Based Queuing (CBQ) (Floyd and Jacobson, 1995) can provide a minimum bandwidth for required flows. The configuration of the schedulers is straightforward from the requirements of the EF flows if we use their bandwidths as the weights in the scheduler (Parekh and Gallager, 1993). 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 variety 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 to 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, do not have any specific quality requirement, there is a lot of flexibility to choose which flows to prioritize. The proposal presented in this paper looks for those flows that make the best use of network resources and produce the highest profit. 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. We manage to find a tradeoff between the total carried traffic and the fairness that offers the best revenue.

Other proposals have focused mainly in routing algorithms with QoS, trying to find the best routes for EF traffic (Chen and Nahrstedt, 1998) (Orda, 1998). 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 (Ma et al., 1996). In this paper we show that even using a shortest-path routing protocol, a substantial improvement can be achieved selecting the optimal bandwidth resources for each BE flow. This bandwidth 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 operating point of the network to a far from optimal situation. We will focus on maximizing carried traffic (and so revenue) for the worst cases of congestion, when the sharing policy becomes critical.

We assume that a flow-based multiplexing and scheduling discipline similar to WFQ is available in each router. The packet scheduler will give priority to EF traffic. The specific reservation for each flow can be selected using some parameters of the scheduling mechanism (the weights). For the best effort traffic we will use precomputed weights that try to select the optimal flows. We 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. We are only specifying how to share the bandwidth among the BE flows.

We use a Linear Programming (LP) approach to calculate these BE weights, trying to maximize the load in the network. This approach has been successfully used in similar flow maximization problems (Chvatal, 1983) (Qiao and Xu, 2002) (Ramaswami and Sivarajan, 1995).

The rest of the paper is organized as follows: in section 1 we present the network scenario. Section 2 explains the maximization objective from the operator point of view. In section 3 we formulate the maximization problem for the traffic carried and present an in-depth analysis of its behavior. Once the maximization of the traffic is studied, section 4 relates this parameter to the maximization of the profit and offers the best configuration for the schedulers. Finally section 5 presents the conclusions that can be drawn from this study.

1. Scenario

The network scenario we study is any topology of nodes (routers) interconnected by links with different bandwidths. Every node is supposed to be in the same administrative domain. If the routing protocol used as IGP (Interior Gateway Protocol) is a link-state protocol like OSPF then the topology information is easy to obtain. Each node participating in the routing protocol has the complete knowledge of the topology from its link-state database. We could also collect this topology information from a central location just by polling one router in the network. Hence, the global information of the topology is known and the configuration of the flows in the nodes is not a coordination problem requiring Service Level Agreements (SLAs) as there is only one administrator in the domain.

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 protocol based on shortest paths (Goldberg, 1993).

An important difference with other optimization works in the literature should be highlighted: *the traffic matrix is not an input parameter*. The traffic matrix is normally a hard to estimate input parameter. The optimization problem we propose does not need this information. It 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. This means that in the case of congestion (the worst case) we ensure the best possible sharing and so the highest profit.

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.

An example topology is shown in figure 1. We will use this specific topology in order to show the behavior of the maximization method proposed. It represents a network with different link bandwidths, several bottlenecks and transient nodes. The number associated with each link is the available bandwidth in units of bandwidth (Mpbs, tens of Mbps, Gbps...) and the transient nodes are filled with a gray pattern. We will extend the study to general network topologies in order to validate the results, but some insights into the optimization method will be better explained with simple topologies like this one.

[Figure 1 about here.]

Each output link in the topology is assumed to be equipped with a packetized version of a Generalized Processor Sharing scheduler (GPS) (Parekh and Gallager, 1993) like WFQ or PGPS. Let $S_i(\tau, t)$ be the amount of session i traffic served in that interval of time and ϕ_i the weights applied to each flow. From (Parekh and Gallager, 1993), for each backlogged session i throughout the time interval $(\tau, t]$, GPS is defined as the scheduling such that equation 1 holds.

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

GPS provides a guaranteed rate for session i of g_i as shown in equation 2, where r is the link bandwidth. Additionally, it provides worst-case network queuing delay guarantees when the sources are constrained by leaky buckets.

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

We are interested in getting the best results from the network even in the worst case. For this reason we will calculate the weights for the BE traffic flows assuming a situation of greedy sources. Every source in the network has as much traffic to send as available bandwidth to every possible non-transient destination in the topology. The resulting configuration for the schedulers will guarantee the carried traffic in this worst case scenario. When the network does not work close to this extreme point, the work-conserving schedulers will guarantee that no traffic is constrained while there is available bandwidth in their paths. Using WFQ schedulers, the sharing of the remaining bandwidth in this situation is proportional to the configured weights.

2. Network operator's objective

The service provider does not have any requirement for best effort traffic, as the name implies. He can select how the sharing among different flows is done, based on private objectives.

The main objective for a private provider is to maximize his profit. This profit depends on the costs for the provider and the prices applied to the users. In this work we will study only the effect of the price for the users and assume that the costs do not depend on the traffic once the network is deployed.

Several studies have shown that "flat" prices for broadband users are unfair (Mackie-Mason and Varian, 1995). These pricing methods usually produce extreme profiles or "heavy users" (Edell and Varaiya, 1999) that generate much more traffic than the average user. Some users with low traffic generation characteristics pay for the bandwidth used by high consumers. This situation can be balanced using a price per bit carried or “usage-based pricing”. We present a study of the maximization of the benefit as a function of the total carried traffic $TotTraff()$ and the price per bit contracted with the user. We define $Cost()$ as the function that gives this price per bit carried. This function will depend on network and configuration parameters. For example, if a minimum bandwidth is guaranteed to every user the cost should be higher the higher this minimum is. Hence, this function could use non trivial expressions and in general the total benefit Π given by equation 3 will not be a linear function.

$$\Pi = TotTraff()Cost() \tag{3}$$

Non-linear maximization problems are much harder to solve than linear ones. In order to avoid a non-linear problem we split the maximization process into two steps. We start by studying the possible ways of maximizing the total carried traffic $TotTraff()$. It is just the addition of the traffic carried in each flow and so it is a linear function. This means that we can use linear optimization techniques on the total traffic carried by the network. In the second stage we will study the interaction with the $Cost()$ function in order to maximize the benefit as the product of both.

3. Methodology for the maximization of carried traffic

In this section we formulate the basic constraints for the Linear Program. This LP will provide the weights for the best effort flows that maximize the total carried traffic.

A Linear Program in standard form follows equations 4a-4c, 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 4b can be generalized. The objective function 4c can be turned into a maximization and the problem can be solved using standard techniques like the well-known Simplex method.

$$\mathbf{A}x = b \tag{4a}$$

$$x \geq 0 \tag{4b}$$

$$\text{minimize } cx \tag{4c}$$

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 stub network router) or several links with other nodes. 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 (equation 5). This path is calculated by a routing protocol and we keep it fixed for our calculations.

$$Path_{s \rightarrow d} = \{(s, n_0), (n_0, n_1)...(n_k, d)\} \tag{5}$$

The amount of traffic in a link $z = (n, m) \in L$ must be limited by the available bandwidth in that link. We denote BW_z the available bandwidth for best effort traffic in a link z once the configuration for the EF flows has been done. If the routing tables are specified, for each link z there is a subset of flows $F \subseteq N \times N$, such that z belongs to the path of every flow in F (equation 6).

$$\forall (s, d) \in F, \quad z \in Path_{s \rightarrow d} \tag{6}$$

All those flows $s \rightarrow d$ use link z and consume bandwidth from BW_z. We express the constraints of limited bandwidth per link in the form of the set of equations 7.

$$\sum_{s,d \in N / z \in Path_{s \to d}} b_{s,d} \leq BW_z, \quad b_{s,d} \geq 0 \tag{7}$$

If N_t is the set of transient nodes, every $b_{s,d}$ with any of the end nodes (s and/or d) in the set of transient nodes must be 0. This requirement is introduced with the constraints in equation 8.

$$b_{s,d} = 0 \quad \forall s, d \in N / s \in N_t \quad \text{and/or} \quad d \in N_t \tag{8}$$

With this basic set of constraints we analyze several optimization problems based on different objective functions for the linear program.

3.1. EVALUATION PARAMETERS

In order to evaluate the configuration provided by the optimization technique we will use several indicators:

– $TotTraff()$ will be the total amount of end to end best effort traffic carried by the network with the configuration obtained from the maximization process. The benefit Π will be directly proportional to the total traffic.

– The minimum bandwidth guaranteed (equation 9). Computed using only the flows that could carry traffic, that means excluding flows from/to transient nodes.

$$minBW = \mathrm{m}in\{b_{s,d}\}, \quad b_{s,d} / s \neq d, s, d \in N - N_t \tag{9}$$

– The disparity D. We define the parameter D as a measurement of the fairness in the bandwidth allocation. From the user point of view, an allocation without preferred flows is fairer than an allocation that reserves more bandwidth to some flows, starving others. We can not offer the same allocation to every user as they use shorter or longer paths with different bottlenecks but we should try to avoid large disparity in the allocation when possible.

  The disparity will be calculated using equation 10. B is the average bandwidth for the BE flows (average of the $b_{s,d}$ between non-transient nodes) and so D is the average difference from the $b_{s,d}$ to B (squared like a variance estimator). This disparity is not an absolute measurement in the sense that we can not use it to compare different topologies, but it is 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 less useful.

$$B^{(\beta)} = \frac{T_{BE}^{(\beta)}}{\|\{(s,d)/s,d \in N - N_t, s \neq d\}\|}$$

$$D = \frac{\sum_{s,d \in N - N_t / s \neq d} (b_{s,d} - B)^2}{\|\{(s,d)/s,d \in N - N_t, s \neq d\}\|} \tag{10}$$

3.2. OBJECTIVE: MAXIMUM CARRIED TRAFFIC

With the set of constraints in equations 7 and 8 we can formulate a linear program. If we want to choose the flows that maximize the amount of traffic carried by the network we only have to solve this problem with an objective function like equation 11.

$$Objective = \max\{ \sum_{s,d \in N / s \neq d} b_{s,d} \} \tag{11}$$

The solution of the Linear Program is the optimal value for each $b_{s,d}$ and the result of the objective function is equal to $TotTraff()$. If we configure in every router the schedulers using these $b_{s,d}$ values as the weights then they are also the amount of bandwidth that each flow will carry in the total congestion case with greedy sources. They are also the minimum bandwidth guaranteed for each flow in any situation.

We call this formulation the *MaxTraffic* methodology. As an example, we apply it to the network in figure 1. In table I we present the evaluation parameters defined in section 3.1 for this topology. As the table shows, the minimum bandwidth assigned to the flows is 0. This means that there are some flows being starved. Some pairs of nodes, in case of network congestion, cannot transfer any amount of traffic while others get bandwidth guaranteed reservations. We look at the specific $b_{s,d}$ and find that 85.45% of the flows have a 0 bandwidth allocation. This is not a reasonable solution, even if the traffic is best-effort, starving completely some flows won't be acceptable from the user point of view.

[Table 1 about here.]

3.3. Objective: Providing a minimum bandwidth and the maximum carried traffic

We can solve the starvation problem exposed in the previous section by forcing a minimum value for each $b_{s,d}$ in the solution of the Linear Program. With this purpose we define an auxiliary variable K. This is the minimum amount of bandwidth assigned to each BE flow. The constraint is expressed in equation 12. We are looking for a solution with a tradeoff between user goals and administrator goals.

$$\forall s, d \in N - N_t \quad b_{s,d} \geq K \geq 0 \tag{12}$$

Now, with the set of constraints from equations 7, 8 and 12 we solve the Linear Program with the objective function in equation 13.

$$Objective = \max\{K\} \tag{13}$$

The solution of this LP provides the maximum bandwidth allocation feasible such that all the valid $b_{s,d}$ (no transient end nodes) are equal. We call this value K_{max}. We could obtain the same result without solving a linear program with a simple algorithm: for each link $z \in L$ compute the number f_z of flows $s \rightarrow d$ such that $z \in Path_{s \rightarrow d}$ and define $M_z = \frac{BW_z}{f_z}$. Then the value of $K_{max} = \min\{M_z\}$.

Once we have computed K_{max} we subtract the bandwidth used by the flows calculated in this first step from BW_z. The new link bandwidth will be $BW'_z = BW_z - K_{max} f_z$. With the network comprising the remaining link bandwidth BW'_z we formulate the goal of maximum network use. Using the same procedure as in section 3.2 we solve the Linear Program that uses the constraints for this new topology (same connectivity but different bandwidth) with the objective function in equation 11. After solving this second LP, the total bandwidth per flow (and so the WFQ weights) is equal to $b_{s,d} + K_{max}$, where $b_{s,d}$ are the solutions for this second Linear Program. The total amount of carried traffic is $\sum_{s,d \in N - N_t / s \neq d} (b_{s,d} + K_{max})$.

We call this formulation the *MinBW* methodology. If we apply it to the example topology in figure 1 we get the results in table II (we also show the results for the *MaxTraffic* methodology for comparison purposes).

[Table 2 about here.]

The minimum bandwidth allocated to the flows is K_{max} as it is calculated in the first step. The total amount of carried traffic has been reduced from the value obtained with the *MaxTraffic* methodology. This is due to the bandwidth that we are allocating to some flows that could be better allocated to other flows, in the sense of obtaining a better maximum traffic. This can be easily seen with the help of figure 2. In this simple topology there are not 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 $TotTraff()$ we count only K we spend $2K$. If the minimum bandwidth constraint does not apply then the flows $1 \rightarrow 2$ and $2 \rightarrow 3$ can be configured with a $2K$ bandwidth and with the same cost in total bandwidth the carried traffic is increased by K. The same procedure can be applied to the flow $3 \rightarrow 1$. This is the reason the solutions with a minimum bandwidth assignment typically do not get to the maximum total carried traffic.

[Figure 2 about here.]

The disparity D has been reduced from the *MaxTraffic* to the *MinBW* methodology. All the flows have now a higher minimum and so they tend to be closer one to each other. As D is a measurement of this distance among the flows, it reduces its value.

While the *MaxTraffic* methodology provides the effective maximum traffic that could be carried by the network, the *MinBW* methodology offers the highest minimum bandwidth for every flow at the expense of a reduction in the carried traffic in the total congestion case. It would be very interesting if we could control the minimum bandwidth allocated in order not to assign the highest possible one, but assign a lower one that results in a higher amount of carried traffic. This means a solution in between those presented in table II, with the flexibility to choose the tradeoff between the carried traffic and the minimum bandwidth guaranteed.

3.4. Objective: Tradeoff between carried traffic and minimum bandwidth

In this section we study the effect of an objective function that combines both objectives of maximizing the carried best effort traffic and obtaining all the traffic assignments larger than 0. This means combining the effects of the methodologies presented in section 3.2 (*MaxTraffic*) and 3.3 (*MinBW*) into a single Linear Program. For this purpose we choose the function in equation 14 as the objective function, where K is the one from equation 12 and α is an independent coefficient that controls the effect of variable K in the problem.

$$Objective = \max\{\sum_{s,d \in N/s \neq d} b_{s,d} + \alpha K\} \quad (14)$$

We expect to control the importance of each part of the objective function using this parameter. The constraints are those in equations 7, 8 and 12. We call this formulation the *Tradeoff* methodology. In the remaining of this section we study the effect of coefficient α on the solution of this Linear Program.

In figure 3 we plot $TotTraff() = \sum_{s,d \in N-N_t/s \neq d} b_{s,d}$ as a function of the weight α and figure 4 shows the minimum bandwidth configured for each flow also as a function of α.

[Figure 3 about here.]

[Figure 4 about here.]

When α is low, K is not as important as $\sum_{s,d \in N-N_t/s \neq d} b_{s,d}$ in the objective function. It is better (in terms of the objective function) to maximize the carried traffic than providing a minimum bandwidth for every flow. That is the reason the Linear Program 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 find the same behavior as with methodology *MaxTraffic*.

When α is high, the minimum bandwidth imposed by K is more important than carrying more flows. The Linear Program tries to get the best minimum assignment and then it will continue maximizing $TotTraff()$. Even with high α, getting higher $b_{s,d}$ we can still improve the result of the objective function. This way, above a certain value of α, we obtain the same behavior as with methodology *MinBW*.

What we are searching with this methodology is a way to find tradeoff solutions that obtain a higher than zero minimum bandwidth by sacrifying some bandwidth but without reserving the maximum K_{max}. If we look at figures 3 (left) and 4 (left) we find a steep transition from one solution to the other instead of a smooth one. This means that for the example topology there is not a tradeoff solution. Even if we look at the steep transition in detail (figures 3 right and 4 right) we do not find the kind of transition we are looking for. The jumps found in this last figures are due to imprecisions in the solving method for the Linear Program when we try to use so much resolution in α.

The reason for this steep transition can be easily explained with the aid of the simple three nodes topology in figure 2. For this topology, the solutions for both extreme methodologies are shown in table III. For low α (or $MaxTraffic$) $Objective(\alpha) = 4 + \alpha 0 = 4$. For high α (or $MinBW$), $Objective(\alpha) = 3 + \alpha 0.5$. While $\alpha < 2$ the solution from *MaxTraffic* obtains a higher result in the objective function and so it is the chosen one. However, when $\alpha > 2$, $Objective(\alpha)_{\alpha>2} > Objective(\alpha)_{\alpha<2}$ and so the solution offered is the one from the $MinBw$ methodology. When $\alpha = 2$ both solutions get the same value in the objective function and so both reach the maximum. This means that when $\alpha = 2$ the linear program could result in any of the two as the solution.

[Table 3 about here.]

From this observation we can compute a lower bound for the value α_t where the transition happens, or the minimum value of α such that the *MinBW* solution provides a higher objective result than the solution from *MaxTraff*. The value that the objective function provides for the solution from the *MaxTraff* methodology is equal to $TotTraff^{MaxTraff}()$. In order for the Linear Program to offer the solution from the *MinBW* methodology instead, it must provide a result of the objective function greater than $TotTraff^{MaxTraff}()$.This value is reached when the term αK raises the total value of the objective function above $TotTraff^{MaxTraff}()$. The value of α in that point, and so the lower bound, is the one given in equation 15.

$$\alpha_t > \alpha_{min} = \frac{TotTraff^{Maxtraffic}() - TotTraff^{MinBW}()}{K_{max}} \tag{15}$$

When $\alpha > \alpha_{min}$, at least the solution from the *MinBW* methodology is better than the solution with $K = 0$ so the linear program will not offer that solution. However, depending on the topology, it could not offer the solution from *MinBW* but a result in between. Figure 5 shows $TotTraff(\alpha)$ for a different topology (not represented in this paper). It offers several intermediate solutions but it does not provide a smooth evolution and those transition points are not easy to locate a-priori. Hence it is not easy to control the tradeoff solution with the parameter α.

[Figure 5 about here.]

The last evaluation parameter that we defined in section 3.1 is the disparity D. We show it as a function of α in figure 6. The figure shows that for low α, below the transition point, we can obtain different solution with different disparity and so different bandwidth allocations. However, in the same range in figures 3 and 4 we can see that the $TotTraff()$ and $minBW$ are the same and so the result of the objective function is the same in this range. We are obtaining different solutions, all of them providing the maximum. Depending on the way the LP is solved and the initial step 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.

[Figure 6 about here.]

The results from this analysis show that we can not tune the parameter α in order to smoothly change the solution of the linear program. We are looking for a slow transition from the maximum traffic option to the maximum minimum traffic method. Instead we get an abrupt change from one to the other. We need a better way to enforce a non zero minimum bandwidth allocation while having some flexibility in terms of the cost in total carried traffic.

3.5. Objective: Maximum carried traffic as a function of the minimum bandwidth guaranteed

In this section we look for solutions in between the *MaxTraffic* and the *MinBW* methodologies. We start using brute force, computing all the tradeoff solutions, but then we show that there is a simple approximation to compute them with a minimum increase in computation cost from the previous methodologies.

We formulate a linear program with the constraints in equations 7, 8 and 12. The objective function is equation 11, the one used in section 3.2 for the *MaxTraffic* methodology. But instead of including the variable K in the objective function we are going to fix it to the value of minimum bandwidth that we want to allocate for the flows. Then, the linear program solves the maximum traffic that could be carried with the best assignment that verifies the constraints.

The valid range for K is $[0, K_{max}]$ with K_{max} as computed in section 3.3. Above K_{max} the program is infeasible. We solve the problem for the example topology and several values of K in the feasible range. Figure 7 (left) shows the smooth transition that we get in the solutions measuring the total carried traffic. We call this function $TotTraff() = TotTrade(K)$. We can now select a non zero minimum bandwidth for the user flows (K) and the figure shows the maximum carried traffic obtainable. Figure 7 (right) shows also how the disparity gets normally reduced as the minimum bandwidth is increased, providing a higher degree of *fairness.*

[Figure 7 about here.]

However, the operator does not know *a-priori* the value of K that he wants. Instead, he could know the maximum reduction in $TotTraff()$ acceptable and based on that he chooses a value for K that provides a carried traffic at least as high as the one needed.

The disadvantage of this methodology is that we are required to solve the linear program for several values of K trying to find the optimal tradeoff point. It would be very interesting if we could know $TotTrade(K)$ without relying to computing many points of the function, because each point implies solving a linear program. In fact, $TotTrade(K)$ in figure 7 could be estimated with a simple straight line. Only two points of the graph are needed in order to know $TotTrade(K)$. These two point could be the two end points of the figure. The first one is the solution when $K = 0$ and so the solution from the *MaxTraffic* methodology. The second one is the solution when $K = K_{max}$ and so the solution from the *MinBW* methodology. This approximation is simpler than computing several points and using minimum squares interpolation to find the best fit first order polynomial. However, we still have to show that this approximation is always good enough.

For this purpose we explore a broad range of topologies using random topology generator techniques. In (Tangmunarunkit and et al., 2002) a comparison of network topology generators is provided: random, structural (hierarchical), and degree based. Random graph generators are the best choice for scenarios like ours, where all the routers are in the same domain or autonomous system. We have chosen the Waxman model (Waxman, 1988) that is a popular method for random graph generation. This method assigns randomly nodes to locations on a plane and the probability that two nodes were connected is a function of the distance d (equation 16), where L is the maximum distance between nodes, $0 < \alpha \leq 1$ is the sensitivity of link formation to distance and $0 < \beta \leq 1$ controls link density (the node's degree).

$$P(d) = \beta e^{\frac{-d}{\alpha L}} \tag{16}$$

We use the BRITE[1] generation tool (Medina and et al., 2001a; Medina and et al., 2001b) in order to create topologies with this model. BRITE is a software developed at the Boston University that uses different methods like Waxman's to generate random topologies. As transient nodes we choose randomly a 40% of the nodes that have at least two links.

We are only interested in the shape of the function $TotTrade(K)$ and how it resembles a straight line. We created hundreds of random topologies and computed $TotTrade(K)$ for each one. Then, we plot a normalized version of the curve in order to just compare the shape among different topologies. In figure 8 we show an example from the analysis carried. We have plotted the result from the topology that gives the best fit to a straight line and the one that gives the worst fit. The figure shows that the first order approximation is not always exact, it depends on the topology. However, the straight line between the extreme points lies always *below* the real $TotTrade(K)$ curves. This means that the line is a worst-case estimation of the total traffic, and most of the time is also very close to the real value.

[1] Boston University Representative Internet Topology Generator

Figure 8 is the result of an uniform distribution of the nodes in a plane. We also tested with a heavy-tailed distribution that creates topologies with clusters of nodes. The results showed an even better fit to a first order approximation.

[Figure 8 about here.]

Hence, solving the Linear Programs for methodologies *MaxTraff* and *MinBW* we can create an analytical expression that approximates fairly well the function $TotTrade(K)$, and so the operating point for the network can be chosen. The expression for this approximation is shown in equation 17, where it holds that the slope of the line is $-m_{traff} = -\alpha_t$ calculated in section 3.4 as the lower bound of the transient point from the *MaxTraffic* solution to the *MinBW* one.

$$\begin{aligned} TotTrade(K) &\approx -\frac{TotTraff^{MaxTraff}() - TotTraff^{MinBW}()}{K_{max}}K + TotTraff^{MaxTraff}() \\ &\approx -m_{traff}K + b_{traff} \end{aligned} \tag{17}$$

4. Network configuration for maximum benefit

From section 3.5 we can estimate the function $TotTrade(K)$ with only the solutions from methodologies *MaxTraffic* and *MinBW*. With this function as input, the network administrator could choose the operating point of the network by selecting the value of K for the users. In this section we propose a decision rule for the K parameter. This rule looks for the maximization of the benefit as it was defined in section 2.

We have defined $Cost()$ as the function that provides the price per bit carried applied to the users. The total benefit obtained is the product $\Pi = TotTraff()Cost()$, where $TotTraff() = TotTrade(K)$. For the design of the $Cost()$ function we should look at the effect that the WFQ weights configured on the network have on the quality experienced by the users.

The value of $b_{s,d}$ (or $b_{s,d} + K$ in the *MinBW* methodology) is the bandwidth allocation for each flow when there is total congestion. However, using WFQ schedulers, when there is no congestion, the higher the $b_{s,d}$ for a flow is the higher the bandwidth share it will get in a link. This means that the user's flows have a higher quality. So the price per bit should be proportional to this weight configuration. However, the maximum allocation achievable depends on the topology of the network and the bottlenecks found in the path of the user's flows. Some users would have higher prices just because the network topology allows them to carry more traffic. Instead, we can apply a price proportional to K. It is the minimum and equal bandwidth allocated to every flow and so it will result in a price proportional to the worst allocation. We do the analysis of the maximization technique using the simple linear function in equation 18, where the cost per bit increases proportional to K and there is an offset value b_{cost}.

$$Cost(K) = m_{cost}K + b_{cost} \tag{18}$$

The estimation of function $TotTrade(K)$ will be represented by equation 17. And the benefit function will take the quadratic form of equation 19

$$\begin{aligned} \Pi(K) &= TotTraff(K)Cost(K) \\ &= -m_{traff}m_{cost}K^2 + \left(b_{traff}m_{cost} - b_{cost}m_{traff}\right)K + b_{traff}b_{cost} \end{aligned} \tag{19}$$

Doing some simple algebra we can solve the optimum value of K that provides the maximum for the benefit function (equations 20 and 21).

$$\frac{\partial \Pi}{\partial K}(K_{opt}) = 0 \quad \Rightarrow \quad K_{opt} = \frac{b_{traff}}{2m_{traff}} - \frac{b_{cost}}{2m_{cost}} \tag{20}$$

$$\Pi(K_{opt}) = \frac{b_{traff}^2}{4m_{traff}} m_{cost} + \frac{b_{cost}^2}{4m_{cost}} m_{traff} - \frac{b_{traff} b_{cost}}{2} \tag{21}$$

Equation 20 shows the dependence of the optimum point on the topology (m_{traff} and b_{traff}) and on the cost function (m_{cost} and b_{cost}). If the parameters of both functions are such that $m_{cost} b_{traff} \gg m_{traff} b_{cost}$ then $K_{opt} \approx \frac{b_{traff}}{2m_{traff}}$ and so *the optimal operating point does not depend on the cost function.* We can obtain this behavior for example if $b_{cost} = 0$, then equations 22 hold. In this situation we can configure the network for maximum benefit, independent of the cost function. This is an important result because it means that as long as the $Cost()$ function takes this form, optimum network configuration can be achieved independently of the price applied to the users.

$$\begin{aligned} K_{opt} &= \frac{b_{traff}}{2m_{traff}} \\ \max_K\{\Pi(K)\} = \Pi(K_{opt}) &= \frac{b_{traff}^2}{4m_{traff}} m_{cost} \end{aligned} \tag{22}$$

In figure 9 we show an example of $TotTrade(K)$, a $Cost(K)$ function and the resulting benefit function for the topology in figure 1. We must note that K_{opt} could lie outside the valid interval for K. Using $b_{cost} = 0$, K_{opt} will be inside the valid interval if $TotTraff^{MinBW}() < \frac{1}{2} TotTraff^{MaxTraf}()$. If the maximum lies outside the valid interval ($K_{opt} > K_{max}$) then the maximum benefit is obtained with $K = K_{max}$. This situation will be found in networks that suffer a low reduction of the total carried traffic when we guarantee some bandwidth to long-path flows.

[Figure 9 about here.]

5. 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 translate 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. We offer a simple procedure for the evaluation of the cost (in terms of traffic carried) of offering a minimum bandwidth for the users. Finally we show that in some situations of rate-based pricing, the optimum network configuration point is independent of the cost per bit applied to the users.

References

Chen, S. and K. Nahrstedt: 1998, '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*.

Chvatal, V.: 1983, *Linear Programming*. NY: Freeman.

Demers, A., S. Keshav, and S. Shenker: 1990, 'Analysis and simulation of a fair queueing algorithm'. *Journal of Internetworking Research and Experience* **1**(1), 3–26.

Edell, R. and P. Varaiya: 1999, 'Providing Internet Access: What we learn from INDEX'. Technical Report, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley.

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

Goldberg, A. V.: 1993, 'Scaling algorithms for the shortest paths problem'. In: *ACM-SIAM Symposium on Discrete Algorithms*. pp. 222–231.

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

Ma, Q., P. Steenkiste, and H. Zang: 1996, 'Routing high-bandwidth traffic in max-min fair share networks'. In: *ACM SIGCOMM96*. pp. 226–217.

Mackie-Mason, J. and H. R. Varian: 1995, 'Some FAQs about Usage-Based Pricing'. *Computer Networks and ISDN Systems* **28**(1-2), 257–165.

Medina, A. and et al.: 2001a, 'BRITE: An Approach to Universal Topology Generation'. In: *International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems- MASCOTS '01*. Cincinnati, Ohio.

Medina, A. and et al.: 2001b, 'BRITE: Boston University Representative Internet Topology Generator'. http://www.cs.bu.edu/brite.

Orda, A.: 1998, 'Routing with end to end QoS guarantees in broadband networks'. In: *IEEE Infocom*. pp. 27–34.

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

Qiao, C. and D. Xu: 2002, 'Distributed Partial Information Management (DPIM) schemes for survivable networks - Part I'. In: *IEEE Infocom*.

Ramaswami, R. and K. Sivarajan: 1995, 'Routing and wavelength assignment in all-optical networks'. *IEEE Transactions on Networking* **3**(5), 489–500.

Tangmunarunkit, H. and et al.: 2002, 'Network Topology Generators: Degree-Based vs. Structural'. In: *ACM SIGCOMM'02*. Pittsburgh, Pennsylvania, USA.

Waxman, B.: 1988, 'Routing of Multipoint Connections'. *IEEE Journal of Selected Areas in Communication* **6**(9), 1617–1622.

White, P.: 1997, 'RSVP and integrated services in the Internet: A tutorial'. *IEEE Communications Magazine* **35**(5), 100–106.

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

Address for Offprints:

Daniel Morató

Universidad Pública de Navarra

Departamento de Automática y Computación

Campus Arrosadia

31006 Pamplona, SPAIN

Tel. +34 948 169854

Fax. +34 948 168924

E-mail: daniel.morato@unavarra.es

List of Figures

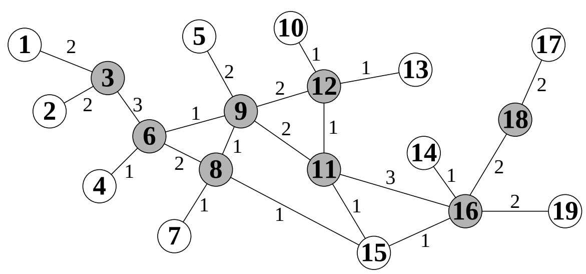

Figure 1. Example scenario

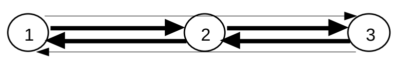

Figure 2. Example topology for the effect of K

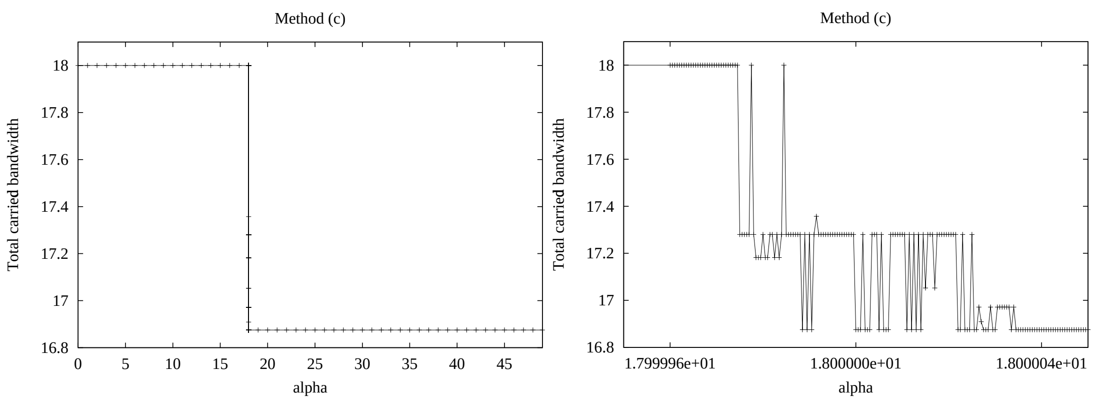

Figure 3. $TotTraff()$ versus the weight α for the example topology

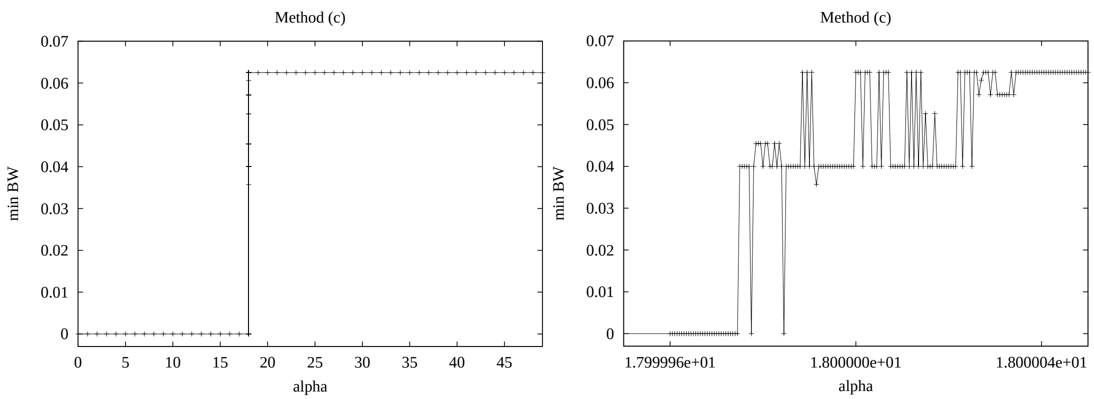

Figure 4. Minimum BW versus the weight α for the example topology

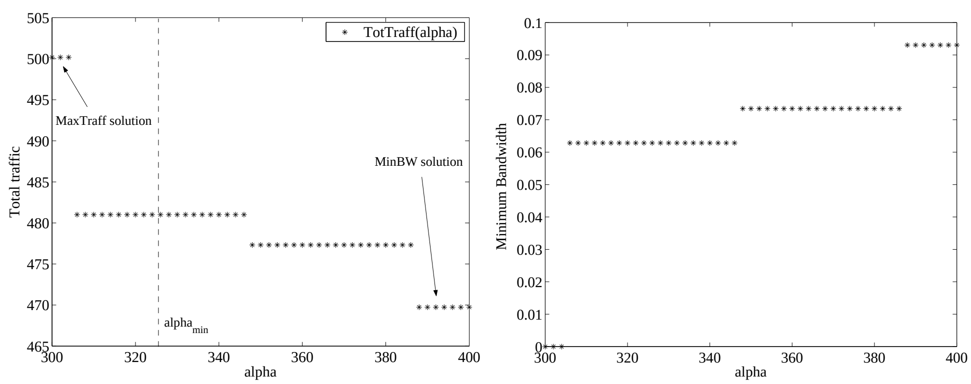

Figure 5. Results for a different topology

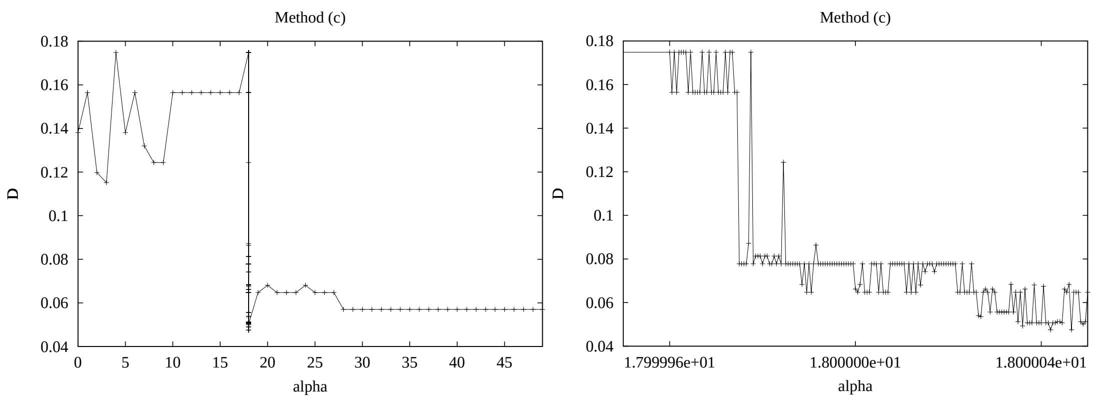

Figure 6. Disparity D versus the weight α for the example topology

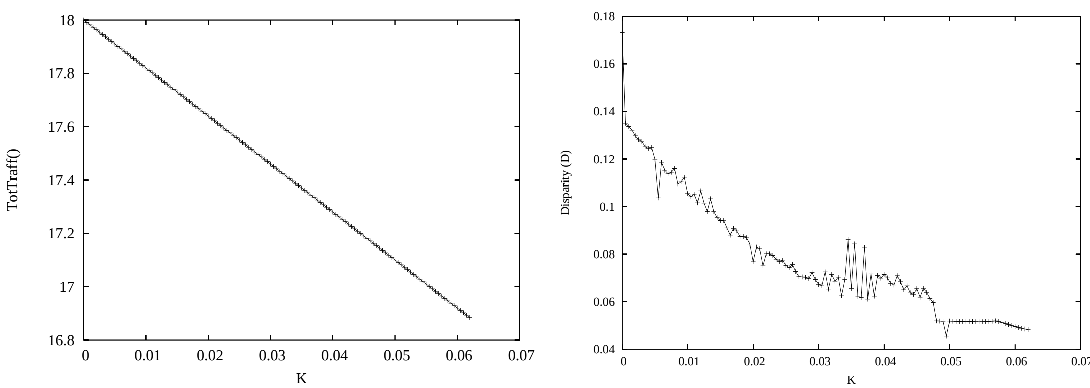

Figure 7. Total traffic and disparity as a function of K for the example topology

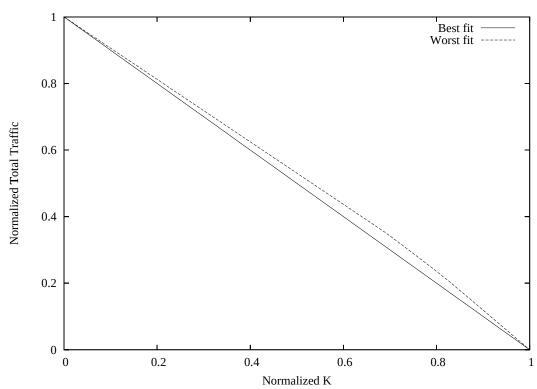

Figure 8. Results of total traffic as a function of K for random topologies

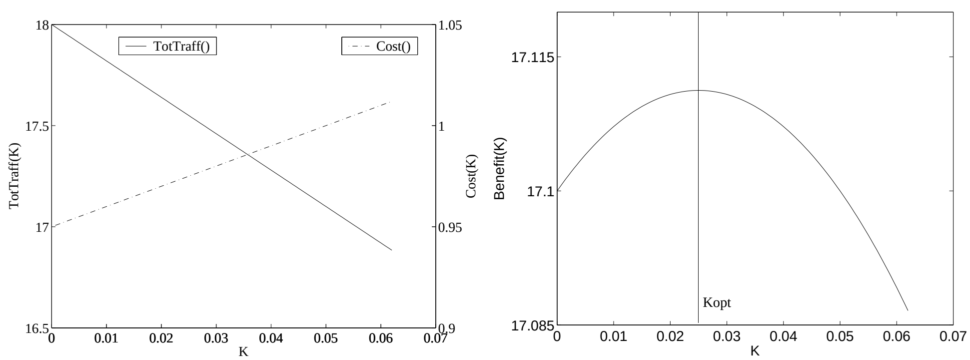

Figure 9. Total traffic and benefit as a function of K for the example topology

List of Tables

Table I. Results for the example topology (*MaxTraffic* methodology)

Methodology	$TotTraffic()$	min BW	D
MaxTraffic	18	0	0.173223

Table II. Results for example topology (*MaxTraffic* and *MinBW* topologies)

Methodology	$TotTraff()$	min BW	D
MaxTraffic	18	0	0.173223
MinBW	16.875	0.0625	0.059207

Table III. Results for a simple topology (*Tradeoff* methodology)

(s, d)	$b_{s,d}^{MaxTraffic}$	$b_{s,d}^{MinBW}$
(1,2)	1	0.5
(2,1)	1	0.5
(2,3)	1	0.5
(3,2)	1	0.5
(1,3)	0	0.5
(3,1)	0	0.5