Convex Optimization of Output Link Scheduling and Active Queue Management in QoS Constrained Packet Switches

Mung Chiang, Bernard L. F. Chan and Stephen Boyd
Electrical Engineering Department, Stanford University, Stanford, CA 94305

***Abstract*— We present two novel algorithms at the ingress and egress of packet switches with QoS provisioning and fairness constraints. We first provide a suite of generalized weighted fair queuing formulations for output link scheduling, where the weights can be dynamically optimized under QoS constraints using the tool of geometric programming. We then provide a suite of active queue management formulations for flexible ingress buffer management, using the tool of semefinite programming. Both sets of formulations are nonlinear, and are special cases of convex optimization problems, which can be solved globally and as efficiently as linear problems.**

I. Introduction

Architectures and algorithms for high performance packet switch with Quality of Service (QoS) provisioning have received much attention in recent years [1]. Incoming packets with different priorities contend for limited resources in both the buffer and the switch fabric. Queuing mechanism, switch scheduling method and link scheduling algorithm used in the switch will therefore affect the QoS parameters of throughput and delay that each packet experiences. In addition to providing preferential treatment to high priority connections, fairness issues must also be taken into account in order to avoid excessive throughput or delay degradation for low priority connections.

This paper focuses on both link scheduling, which is usually used at the egress, and active queue management, which is usually implemented at the ingress. By using the powerful computational tools of convex optimization, in particular, geometric programming and semidefinite programming, we provide a suite of formulations that efficiently find the optimal tradeoff between prioritized treatment and fairness constraints. Although these formulations are nonlinear problems, they belong to the class of convex optimization and can be solved in polynomial time.

II. Convex Optimization, Geometric Programming and Semidefinite Programming

Convex optimization [2] refers to minimizing a convex objective function subject to a convex constraint set. Solving convex optimization problems is easy. It is easy in theory because a local minimum is a global minimum and the duality gap is zero. It is also easy in practice because there are efficient algorithms, such as the interior point method [3], that find the globally optimal solution as fast as linear programming. However, identifying and formulating convex optimization problems may not be straight forward.

The particular type of convex optimization we use for Generalized Weighted Fair Queuing in section III is in the form of geometric programming. We first introduce the following

Definition 1: A monomial is a function $f : \mathcal{R}^n \rightarrow \mathcal{R}$, where the domain contains all real vectors with positive components:

$$f(x) = c x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}, \;\; c \geq 0 \; and \; a_i \in \mathcal{R} \qquad (1)$$

Definition 2: A posynomial is a sum of monomials $f(x) = \sum_k c_k x_1^{a_{1k}} x_2^{a_{2k}} \cdots x_n^{a_{nk}}$.

For example, $f(x) = x_1^e x_2^{-\sqrt{2}}$ is a monomial and $f(x) = x_1^2 x_3 + 2 x_1^{\pi} x_2^{-1}$ is a posynomial. Geometric programming is an optimization problem in the following form:

$$\begin{array}{lccc} minimize & f_0(x) & & \\ subject\ to & f_i(x) & \leq & 1 \\ & h_j(x) & = & 1 \end{array} \qquad (2)$$

where f_0 and f_i are posynomials and h_j are monomials. Geometric programming in the above form is not a convex optimization problem. However, with a change of variables: $y_i = \log x_i$ and $b_{ik} = \log c_{ik}$, we can put it into convex form:

$$\begin{array}{lcclcc} minimize & p_0(y) & = & \log \sum_k \exp(a_{0k}^T y + b_{0k}) & & \\ subject\ to & p_i(y) & = & \log \sum_k \exp(a_{ik}^T y + b_{ik}) & \leq & 0 \\ & q_j(y) & = & a_j^T y + b_j & = & 0 \end{array} \qquad (3)$$

It can be verified that the log of sum of exponentials is a convex function. Therefore, p_i are convex functions and q_j are affine functions, and we have a convex optimization problem. Note that if all posynomials are in fact monomials, geometric programming becomes linear programming.

Geometric programming has been used in different engineering applications. Recent examples include CMOS op-amp design [4] and resource allocation in wireless networks [5].

The particular type of convex optimization we use for active queue management in section IV is semidefinite programming. When inequality constraints of an optimization problem is generalized to inequalities induced by the convex positive semidefinite cone, we have the following semidefinite program (SDP):

$$\begin{array}{ll} minimize & c^T x \\ subject\ to & x_1 F_1 + \cdots + x_n F_n + G 0 \succeq \\ & Ax = b \end{array} \quad (4)$$

where $F_i = F_i^T \in \mathcal{R}^{k \times k}$, and the inequality constraint will be called linear matrix inequality, with $A \succeq 0$ denoting that A is a positive semidefinite matrix. There are some common methods to be used to convert our problem at hand to a semidefinite program. They include converting a linear inequality into a linear matrix inequality with diagonal matrices, introducing a dummy variable for conversion to epigraph form, and using Schur's complement [2].

The running time of convex optimization usually scales with the logarithm of the problem size, and for most practical applications, converge in 5 to 10 iterations. We can also use three types of heuristics that simplify the computation even further:

1) Use the structure of the problem to simplify the algorithmic steps.
2) Simplify the problem input data structure.
3) Perform incremental update of the current optimal solutions whenever input data changes in real time applications.

III. Generalized weighted fair queuing

A. Introduction

Weighted fair queuing [6] (WFQ) is a commonly used technique to strike a proper balance of resource allocation with prescribed fairness parameters. It is particularly widely used in egress link scheduling algorithms in many packet switch architectures. However, there is a major limitation on weighted fair queuing methods, its fairness parameters cannot be dynamically optimized in a computationally efficient way.

By using geometric programming, we provide a method to optimize the fairness parameter for a particular connection under a variety of constraints on fairness parameters for other connections. The objective can also be optimizing for a weighted sum of connections with non-integer weights, or for the worst performing connection. Such formulations can accommodate a large number of variables for dynamic and constrained optimization of WFQ parameters.

We first give a briefly overview of the Generalized Processor Sharing (GPS) scheme, which WFQ approximates on a packet by packet base. Suppose there are N connections trying to share a resource with a total rate of r. For example, there are 16 virtual output ports sharing one egress linecard with a transmission rate of r packets per second. Assign a positive real number ϕ_i to connection i. This number prescribes the QoS parameter for the connection. Note that ϕ_i are fixed parameters. Share the resource according to the following formula:

$$\frac{S_i(t_1, t_2)}{S_j(t_1, t_2)} \geq \frac{\phi_i}{\phi_j} \quad (5)$$

where $S_i(t_1, t_2)$ is the amount of resources allocated to connection i from time t_1 to t_2.

It has been be shown that such a GPS algorithm would guarantee that the rate g_i received by connection i is bounded by

$$g_i = \frac{\phi_i}{\sum_{j=1}^{N} \phi_j} r \quad (6)$$

The ideal GPS assumes infinite divisibility of data flow. WFQ approximates the ideal GPS and assumes that a packet cannot be further divided in scheduling. In this section, we present a flexible and efficient way to optimize the fairness parameters ϕ_i for GPS, and the packet based approximation in WFQ is straight forward.

B. Convex Optimization Formulations

In this subsection, we show formulations that optimize the performance of weighted fair queuing subject to fairness constraints. We call this the Generalized Weighted Fair Queuing.

Suppose we would like to maximize the rate of a particular connection i^*, subject to fairness constraints on the rates for other connections, by varying the QoS parameters ϕ_i. Although this is a nonlinear problem, we can turn this problem into a geometric programming problem, which is as easy to solve for global optimality as a linear problem.

Proposition 1: The following Generalized Weighted Fair Queuing formulation is a convex optimization problem with WFQ parameters ϕ_i as the optimization variables.

$$\begin{array}{lll} maximize & g_{i^*} & \\ subject\ to & g_i \geq b_i, & i \neq i^* \end{array} \quad (7)$$

where b_i are the QoS constraint constants that lower bound the provisioned rate each connection receives. The proof is rather straight forward after rewriting the objective as minimization of $\frac{1}{g_i^*}$ and the constraints as $\frac{1}{g_i} \leq \frac{1}{b_i}$. It can be verifed that both the objective function and the constraints are posynomials of the variables ϕ_i.

In fact, we can maximize not just for one connection, but a weighted sum of many connections in a set I to achieve proportional fairness, or the worst case connection to achieve min-max fairness. For proportional fairness, it would be desirable to jointly optimize ϕ_i and weights w_i. Indeed, the fairness weights in the proportional fairness algorithm can become problem variables and be allowed to be non-integers, and while still maintaining the geometric program form.

Proposition 2: The following variations of Generalized Weighted Fair Queuing are convex optimization problems with

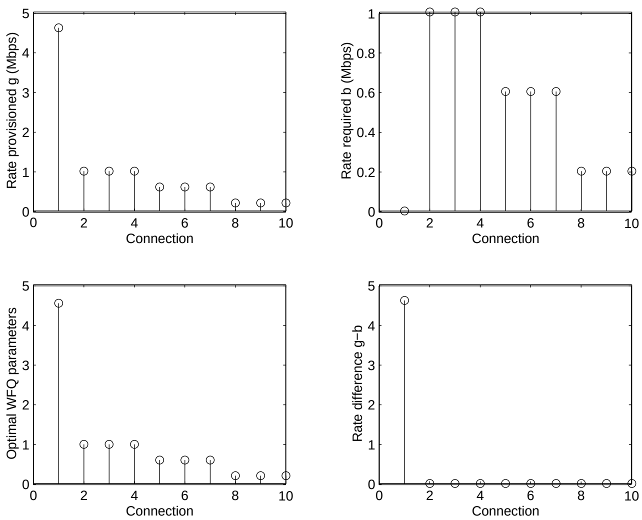

Fig. 1. Simulation 1 for Generalized WFQ

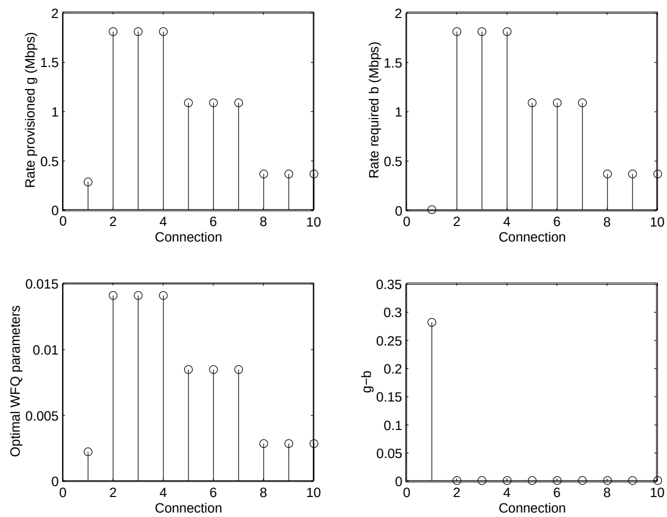

Fig. 2. Simulation 2 for Generalized WFQ

the WFQ parameters ϕ_i and the fairness weights w_i as the optimization variables.

$$\begin{array}{lll} maximize & \sum_i w_i g_i, \;\; i \in I & \\ subject\ to & g_i \;\; \geq \;\; b_i & \\ & w_i \;\; \geq \;\; w_{i,lb} & \\ & w_i \;\; \leq \;\; w_{i,ub} & \end{array} \quad (8)$$

where $w_{i,lb}$ and $w_{i,ub}$ are constants of upper bounds and lower bounds for fairness weights w_i, respectively.

Proposition 3: The following variations of Generalized Weighted Fair Queuing are convex optimization problems with the WFQ parameters ϕ_i as the optimization variables.

$$\begin{array}{ll} maximize & \min_i g_i \\ subject\ to & g_i \;\; \geq \;\; b_i \end{array} \quad (9)$$

The proofs of Propositions 2 and 3 follow the same outline as the proof of Proposition 1. Depending on how often the fairness parameters need to be optimized, we can use the computationally efficient primal dual interior point algorithms [3] for infrequent parameter optimization, or even simpler heuristics through approximations in algorithm and problem setup for frequent parameter optimization. Incremental parameter update with minimal computational load can also be supported. Details of the simplified heuristics are developed in [8].

C. Simulations

We present two numerical examples on the basic version of Generalized Weighted Fair Queuing as in Proposition 1. In the first example shown in Figure 1, we have an egress link of a total rate of $10M\,bps$ to be shared by 10 connections with different QoS requirements. We would like to optimize for connection 1, subject to constraints for all the other connections. The resulted rates g_i through geometric programming is shown together with the optimal values of the variables ϕ_i, and the required rates b_i. As can be seen from the difference between rates allocated g_i and rates required b_i, the convex optimization results satisfy the QoS constraints for all connections while finding the globally optimal set of ϕ_i to maximize the throughput for connection 1.

Another example is shown in Figure 2. The simulation setup is similar to that in example 1, except that the QoS constraints for all connections are more stringent. Therefore, in order to meet these constraints, the optimized rate for connection 1 is much smaller than that in example 1. This shows the effect of constraint tightness on the optimal value of the objective function. A more complete picture of the slackness of the constraints can be shown through the dual problem of geometric programming, which has an interesting physical interpretation [5].

IV. Flexible active queue management

A. Introduction

During congestion, buffer overflow will occur and tail drop will take place. In order to prevent the undesriable and unfair effects of tail drop, we need to implement active queue management and systematically drop packets. The basic version current used are RED [7] and its variations. In this section, we provide a new algorithm with much flexibility using the tool of semidefinite programming.

At a particular time slot, we are given a non-negative matrix A that represents the switching to be done in a switch with N input ports and N output ports. For a given routing table and incoming traffic, A_{ij} is the proportion of traffic coming in at input port i that is intended for and needs to be routed to output port j. It is easy to verify that $\sum_j A_{ij} = 1$, and therefore A is a stochastic matrix. Let z be the incoming traffic vector, where

z_i is the amount of ingress traffic coming to input port i of the switch. The amount of egress traffic at output ports is therefore $y = Az$. Therefore, egress port traffic is the image of the ingress port traffic under the linear transformation represented by the stochastic matrix A.

Ideally, the given switching matrix should be a permutation matrix and have only one positive entry in each row, which implies that packets from only one input port needs to be switched to each output port and no contention is resulted. However, this condition does not hold in general, and contention avoidance needs to be done by active queue management during congestion.

Our task is to design a queue management matrix B that acts as a filter before passing on the traffic to the scheduling algorithm in the switching fabric. B is designed based on the given switching matrix A so as to minimize potential contention under QoS constraints, where B_{ij} is the actual proportion of signals coming into input port i that will be switched to output port j. In general, $B \neq A$ because A just arises from the given traffic pattern that may not represent the best tradeoff between collision avoidance and active queue management.

The variables of this optimization are B_{ij}, the entries of B. The intuition behind the algorithm is that by decreasing some entries of B and increasing other entries, we are shaping the incoming traffic. Some streams of traffic might have a lower priority and yet they have more packets to be switched than those with higher priority. Therefore, we would increase the weight of those higher priority streams in the switching matrix and decrease the weight of the lower priority streams. We hope that we can also reduce the number of packets to be switched to the output ports after this shaping.

B. Convex Optimization Formulations

The basic problem formulation is as follows:

$$\begin{array}{lll} minimize \| B\| & & \\ subject\ to\ B_{ij} & \geq & 0 \\ \sum_j B_{ij} & \leq & 1 \\ B_{ij} & \geq & K_{ij}A_{ij},\ i \neq j \end{array} \qquad (10)$$

where $\|B\|$ is the spectral norm of B and K_{ij} represent the fairness constraints. We now first explain the objective function of the above optimization problem in terms of eigenvalue analysis and induced matrix norms, followed by an explanation of the QoS constraints.

In general, we would like the size of B, properly measured by some metrics, be small so that the aggregate potential contention is reduced. There are two reasons why $\|B\|$ is a good measure of potential collision at the output ports, and therefore an appropriate objective function to be minimized. First, let $C = B^t B$. By the definition of the spectral norm of a matrix, $\|B\| = \sqrt{\sigma_{max}}$, where σ_{max} is the largest eigenvalue of C. Each entry in C represents the amount of collision between any two input ports $I(i)$ and $I(j)$ on all output ports $O(k)$:

$$C_{ij} = \sum_k (I(i) \to O(k))(I(j) \to O(k)).$$

Since all traffic vectors z coming into the switch can be decomposed into a linear combination of eigenvectors of C, minimizing σ_{max} is equivalent to minimizing the worst mode of total contention caused by all possible traffic vectors z.

Another intrepretation stems from the fact that the spectral norm is an induced norm. Specifically, there is a corresponding vector norm $|z|$ such that $\|B\| = \max_{|z|=1} |Bz|$. A minimax intrepretation of the objective function in problem (10) follows:

$$\min_B \max_{|z|=1} |Bz|$$

In this minimax interpretation, the incoming traffic chooses the worst case for contention, and we design the switching matrix B so as to avoid contention under the worst case traffic.

As for the constraints, the constants $K_{ij} \geq 0$ represents the fairness weight attached to the traffic stream from input i to output j. Larger K_{ij} implies higher priority for the connection from input port i to output port j. If allowed by the fairness constraints, some proportion of the traffic at an input port can be dropped with a certain probability, so as to minimize potential contention in the current time slot,hence achieving the goals of active queue management. Therefore, we have the substochastic constraint $\sum_j B_{ij} \leq 1$.

This problem can be put into the form of a semidefinite program, with efficient algorithms for finding the globally optimal solutions.

Proposition 4: The optimization problem (10) is a SDP and therefore a convex optimization problem.

Proof We prove the above proposition by converting the optimization problem in problem (10) into the standard SDP form. We introduce a dummy variable t, and show that problem (10) is equivalent to the following epigraph form

$$\begin{array}{lll} minimize\ t & & \\ subject\ to\ \|B\| & \leq & t \\ t & \geq & 0 \\ B_{ij} & \geq & 0 \\ \sum_j B_{ij} & \leq & 1 \\ B_{ij} & \geq & K_{ij}A_{ij},\ i \neq j \end{array} \qquad (11)$$

which, by Schur's complement, is equivalent to

$$\begin{array}{lll} minimize\ t & & \\ subject\ to \begin{pmatrix} tI & B \\ B^T & tI \end{pmatrix} & \succeq & 0 \\ t & \geq & 0 \\ B_{ij} & \geq & 0 \\ \sum_j B_{ij} & \leq & 1 \\ B_{ij} & \geq & K_{ij}A_{ij},\ i \neq j \end{array} \qquad (12)$$

This is in the standard form of SDP with the first constraint being the linear matrix inequality constraint. Now fast algorithms [3] can be used to solve this convex optimization problem.

C. Extensions

We briefly mention two extensions of the basic formulation in the previous subsection. First, we sometimes are interested in the dual problem of optimizing the QoS parameters subject to a size constraint on matrix C. Through the use of convex optimization and its heuristics, we can indeed maximize the fairness constraint K_{ij} for a pair of high priority connection across the switch fabric, subject to a constraint on the total potential collision and fairness constraints on all other K_{ij}.

Second, apart from minimizing the spectral norm of B, we can use any other induced norms for B as the measure of the size of B and the objective to be minimized.

Proposition 5: Any induced matrix norm B is a convex function of B.

Proof. The induced norm [2]

$$\|B\|_{a,b} = \sup_{z \neq 0} \frac{|Bz|_a}{|z|_b}$$

can be expressed as a function

$$f(B) = \sup\{x^t Bz : |x|_{a^*} \leq 1, \|z\| \leq 1\}$$

where $|x|_{a^*}$ is the dual norm of $|x|_a$. Since f is a supreme of a family of convex functions, it is also a convex function.

Therefore we can replace the spectral norm in problem (10) by any other induced matrix norm and the minimax interpretation follows. Details of these examples and extensions can be found in [8].

D. The Worst Queue Management Matrix

In this subsection, we prove a related result, that convex optimization can also be used to find the worst case queue management matrix B, if we measure the size of B by the product of all its singular values, or equivalently, the product of all the eigvevalues of the potential collision matrix $C = B^t B$. By making the assumptions that $\sum_i B_{ij} \leq 1$ (putting an upper bound on the total potential traffic each output port can handle), we can maximize the product of all eigenvalue of C as follows.

$$\begin{array}{lll} maximize \prod & \sigma_i(B^t B) & \\ subject\ to\ B_{ij} & \geq & 0 \\ \sum_j B_{ij} & \leq & 1 \\ \sum_i B_{ij} & \leq & 1 \\ B^t B & \succ & 0 \\ B_{ij} & \geq & K_{ij} A_{ij},\ i \neq j \end{array} \qquad (13)$$

Proposition 6: Problem (13) is a convex optimization problem.

The proof is outlined as follows. Since $\sum_j C_{ij} = \sum_j \sum_k B_{ik} B_{jk} \leq 1$, it can be shown from Perron-Frobenious theory of non-negative matrices that $1 > \det(C) > 0$, and that maximizing $\prod \sigma_i(C)$ is equivalent to minimizing $-\log\det(C)$. Furthermore, the second derivative of $g(t) = -\log\det(Z + tV)$ evaluated at 0 is non-negative, and $-\log\det(C)$ is therefore a convex function of C. Proposition 6 follows.

V. Conclusions

We show that the computationally efficient tools of convex optimization can be effectively used in designing QoS and fairness constrained algorithms for high performance packet switches. In particular, we provide a Generalized Weighted Fair Queuing for output link scheduling, as well as several proportional and minmax fairness extensions, through geometric programming. We also provide a flexible active queue management formulation, and prove that it can be transformed into semidefinite programming problems. Although these formulations are nonlinear problems, globally optimal solutions efficiently found through convex optimization algorithms.

References

[1] *IEEE Communications Magazine* December 2000.
[2] S. Boyd and L. Vandenberghe, *Convex Optimization and Engineering Applications,* Stanford University EE 364 Course Reader 1999.
[3] Yu. Nesterov and A. Nemirovsky, *Interior Point Polynomial Method in Convex Programming,* SIAM 1994.
[4] M. del Mar Hershenson, S. Boyd and T. H. Lee, 'Optimal design of a CMOS op-amp via geometric programming,' *IEEE Transactions on Computer Aided Design,* March 2001.
[5] D. Julian, M. Chiang and D. O'Neill, 'Resource allocation for Constrained QoS Provisioning in Wireless Cellular and Ad Hoc Networks,' *Proc. IEEE INFOCOM,* May 2002.
[6] A. K. Parekh and R. Gallager, 'A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case,' *IEEE Transactions on Networking,* June 1993.
[7] S. Floyd and V. Jacobson, 'Random Early Detection Gateways for Congestion Avoidance,' *IEEE Transaction of Networking,* August 1993.
[8] M. Chiang, B. L. F. Chan and S. Boyd, 'Applications of Convex Optimization in Packet Switch Design,' under preparation.