Rate Allocation under Network End-to-End Quality-of-Service Requirements

Mohamed Saad, Alberto Leon-Garcia and Wei Yu
Dept. of Electrical and Computer Engineering
University of Toronto, Canada
{saadme,leongarcia,weiyu}@comm.utoronto.ca.ca

***Abstract*— We address the problem of allocating transmission rates to a set of network sessions with end-to-end bandwidth and delay requirements. We give a unified convex programming formulation that captures both average and probabilistic delay requirements. Moreover, we present a distributed algorithm and establish its convergence to the global optimum of the overall rate allocation problem. In our algorithm, *session sources* update their rates as to maximize their individual benefit (utility minus bandwidth cost), the *network* partitions end-to-end delay requirements into local per-link delays, and the *links* adjust their prices to coordinate the sources' and network's decisions, respectively. This algorithm relies on a network utility maximization approach, and can be viewed as a generalization of TCP and queue management algorithms to handle end-to-end QoS. We also extend our results to deterministic delay requirements when nodes employ Packet-level Generalized Processor Sharing (PGPS) schedulers.**

I. Introduction

Next generation networks will evolve from delivering best-effort traffic to supporting applications with different quality-of-service (QoS) requirements, e.g., bounded end-to-end delay, bounded packet loss and minimum guaranteed rate. Examples of such applications are voice, streaming multimedia, gaming, distributed computing and remote surgery. QoS requirements are usually negotiated at the session establishment time. Once a session is admitted, its QoS requirements must be adhered to by the network.

This paper considers the following rate allocation problem that is fundamental in providing QoS. We are given a set of communication sessions, each characterized by its route, a utility function and a set of QoS requirements. The term "session" is used somewhat broadly in this paper. In particular, a session could be a virtual circuit (as in ATM), or an entire packet flow originating at one node and destined for another node (as in MPLS or IntServ based IP networks). The goal is to find a set of session data rates, such that the desired levels of QoS are achieved, and the total network utility (i.e., the social welfare) is maximized. We simply refer to this problem as *QoS rate allocation*.

This paper considers minimum rate (bandwidth) and end-to-end delay as the QoS measures. In particular, we may require a bounded *average* end-to-end delay. This *soft* QoS guarantee is useful for *predictive* as opposed to *guaranteed* services, e.g., FTP and HTTP. We may also allow the end-to-end delay of a session to violate its required limit with some *probability*. Finally, we may require a bounded end-to-end deterministic (worst-case) session delay. Note that deterministic and probabilistic delay guarantees are useful for voice, streaming multimedia and gaming.

Problems of allocating network resources to provide QoS have received considerable attention. The studies in [7] and [24] address the problem of allocating rates on the links of a *single* path (or multicast tree), such that an end-to-end delay requirement is satisfied. In particular, [24] presents a technique to map end-to-end QoS requirements directly into link rates. The study in [7], however, presents a framework in which end-to-end QoS requirements are first partitioned into local per-link QoS requirements, then these local QoS requirements are mapped into link rates that are reserved along the path (or multicast tree). By considering only *one* session, these studies have overlooked the intrinsic difficulty of resolving contention of multiple co-existing sessions for bandwidth.

Some studies, e.g., [15] and [9], considered the QoS partitioning problem separately. Given a *single* path (or multicast tree) and an additive end-to-end QoS (e.g., delay) budget, the study in [15] addresses the problem of partitioning the end-to-end QoS budget into local per-link QoS requirements, such that a given cost function is minimized. This study assumes the availability of a per-link cost function that accurately captures the global optimization objective. It is straightforward to see that this assumption is somewhat unrealistic [5]. The study in [9], however, addressed the same QoS partitioning problem as to maintain explicit load-balance among links.

In this paper, we focus on resource allocation, and assume that a QoS routing algorithm has already been used to define the sessions' paths. QoS routing (see, e.g., [22]) is concerned with finding a path (or multicast tree) that optimizes a certain design objective while ensuring that some QoS constraints are met. Some studies (e.g., [7]) argue that practical algorithms should address routing and resource reservation separately.

The rate allocation problem considered in this paper is related to the basic network utility maximization (NUM) problem introduced by Kelly *et. al.* in [12] and [13]. Given a network and a set of sessions, the basic NUM problem is concerned with finding a set of session rates that maximizes the overall network utility without violating link capacity constraints. A standard algorithm to solve this problem is based on Lagrangian decomposition (see, e.g., [17]). Several subsequent studies, e.g., [17] and [19], address rate allocation problems using the NUM framework. Moreover, existing TCP and queue management algorithms can be interpreted as implicitly solving a NUM problem [16], [18]. All these studies, however, consider rate allocation in a best-effort network.

A few recent studies that follow the utility/revenue maximization framework for QoS traffic do exist. The study in [14] considers constraints on loss, maximum delay and blocking. The authors formulate the rate allocation problem as a nonlinear integer program that is intractable to solve. The study in [11] considers the allocation of bandwidth and buffer space to sessions with constraints on loss and maximum delay. The problem formulation turns out to be non-convex if delay constraints are considered. Consequently, typical optimization algorithms may not converge to the global optimum [2].

The *contribution* of this paper is threefold.

- We give (in Section II) a unified convex programming formulation for the QoS rate allocation problem that captures both average and probabilistic delay requirements, in addition to minimum bandwidth requirements.
- We present (in Section III) a distributed algorithm and establish its convergence to the global optimum of the overall rate allocation problem. In our algorithm, *session sources* update their rates as to maximize their individual benefit (utility minus bandwidth cost), the *network* partitions end-to-end delay requirements into local per-link delays, and the *links* adjust their prices to coordinate the sources' and network's decisions, respectively. This algorithm relies on a network utility maximization approach, and can be viewed as a generalization of TCP and queue management algorithms (as interpreted by Low [16]) to handle end-to-end QoS.
- We extend (in Section IV) our results to deterministic delay requirements when nodes employ Packet-level Generalized Processor Sharing (PGPS) [20] schedulers.

Our notations are fairly standard. For any random variable T, we use $E(T)$ to signify the expected value of T. For any event A, we use $P(A)$ to signify the probability that A occurs. For any real number x, we let $[x]^+ = \max\{x, 0\}$.

II. Problem Definition

A network is modeled as a set L of links. Associated with every link $l \in L$ is a finite capacity c_l (bits/s). The set of links are shared by a set S of communication sessions. Every session $s \in S$ is characterized by the following attributes:

1) A route that consists of the subset $L(s) \subseteq L$ of links.
2) An end-to-end (average, probabilistic or deterministic) delay requirement D_s (seconds).
3) A minimum rate (bandwidth) requirement R_s (bits/s). Throughout, we use the notions rate and bandwidth interchangeably.
4) A utility function $U_s(x_s)$, where x_s is the transmission rate of session $s \in S$.

We assume that the functions $U_s(x_s)$ are nondecreasing and strictly concave in x_s. Utility functions with these characteristics are commonly used in the rate allocation and pricing literature (see, e.g., [13]). For example, $U_s(x_s) = \log(x_s)$ is nondecreasing and strictly concave. Moreover, maximizing $\sum_s \log(x_s)$ ensures proportional fairness among sessions [12]. In the sequel, we assume that $U_s(x_s)$ is *only known* by session s, and unknown to all other sessions in the network. We also assume that the buffers at the network nodes are large enough to allow us to neglect packet loss.

In this section (and Section III) we assume a *general packet length distribution* with mean $1/\mu$ and variance σ^2. If every link is modeled as an $M/G/1$ queue, the the average delay on link l is given by (see, e.g., [8]):

$$E(T_l) = \frac{(1-\beta)/\mu}{c_l} + \frac{\beta/\mu}{c_l - \sum_{s:l\in L(s)} x_s}, \tag{1}$$

where β is a *constant* given by:

$$\beta = (1 + \mu^2\sigma^2)/2. \tag{2}$$

Note that $\sum_{s:l\in L(s)} x_s$ is the total flow of sessions crossing link l. It is not difficult to verify that (1) is equivalent to the well-known *Pollaczek-Khinchin* (*P-K*) *formula* [3] for average delay in $M/G/1$ queues. By Kleinrock's independence assumption, the $M/G/1$ queue is a good approximation for the behavior of individual links for networks involving Poisson arrivals at entry points, a densely connected network and moderate-to-heavy traffic loads [3].

The following observation indicates that the capacity constraint $\sum_{s:l\in L(s)} x_s \le c_l$ is implied by the delay constraint $E(T_l) \ge \frac{1}{\mu c_l}$. The proof is omitted due to space limitations.

Observation 1: Let $E(T_l)$ be given by (1). Then, $0 \le \sum_{s:l\in L(s)} x_s \le c_l$ if and only if $E(T_l) \ge \frac{1}{\mu c_l}$.

We formulate the QoS rate allocation problem as follows:

$$\begin{aligned}
\text{maximize} \quad & \sum_s U_s(x_s) \\
\text{subject to} \quad & \sum_{s:l\in L(s)} x_s \le c_l - \frac{\beta}{\mu d_l - \frac{1-\beta}{c_l}}, \ \forall l && \text{(3a)}\\
& \sum_{l\in L(s)} d_l \le p_s D_s, \ \forall s && \text{(3b)}\\
& x_s \ge R_s, \ \forall s && \text{(3c)}\\
& d_l \ge \frac{1}{\mu c_l}, \ \forall l, && \text{(3d)}
\end{aligned}$$

where the *variables* are:

x_s	transmission rate of session $s \in S$ (in bits/s);
d_l	local average delay allowed on link $l \in L$ (in seconds).

The objective in this formulation is to maximize the overall network utility. It is straightforward to see that (3a) is equivalent to $E(T_l) \le d_l$, where $E(T_l)$ is given by (1). Thus, (3a) ensures that the average delay on every link l is guaranteed not to exceed d_l. Inequality (3b) ensures that $\frac{\sum_{l\in L(s)} E(T_l)}{D_s} \le p_s$. By Markov's inequality, this ensures that $P(\sum_{l\in L(s)} T_l \ge D_s) \le p_s$. In other words, (3b) guarantees that the probability of violating the end-to-end delay requirement D_s is bounded by p_s. Note that if $p_s = 1$, (3b) will ensure that $\sum_{l\in L(s)} E(T_l) \le D_s$, i.e., that the average end-to-end delay of s is bounded by D_s. In short, (3b) captures both *average* and *probabilistic* delay requirements. Constraint set (3c) ensures that all minimum rate requirements are satisfied, i.e., every session s will be able to transmit at a rate of at least R_s. Finally, (3c) and Observation 1 imply that the total flow on link l does not exceed its capacity.

Now, we make the following observation.

Observation 2: If the utility function $U_s(x_s)$ of every session s is concave in the session rate x_s, then the QoS rate allocation problem as given by (3) is a convex optimization problem.

Proof: Maximizing a concave objective function subject to constraints of the form $f(x) \leq 0$, where $f(x)$ is a convex function of the variable set x, is a convex optimization problem [4]. Constraints (3b)-(3d) are convex because they are linear. Moreover, it can be verfified that $\sum_{s:l\in L(s)} x_s - c_l + \beta / \left(\mu d_l - \frac{1-\beta}{c_l}\right)$ is a convex function in the variables $\{x_s : s \in S\}$ and $\{d_l : l \in L\}$ as long as $\mu d_l \geq (1-\beta)/c_l$ (which is ensured by (3d) and the fact that β is nonnegative). Therefore, constraint (3a) is also convex. Consequently, (3) is a convex optimization problem as long as, for every s, $U_s(x_s)$ is a concave function of x_s. ■

In general, the QoS rate allocation problem (3) may not have a feasible solution. In other words, there may not exist a set session rates that satisfy constraints (3a)-(3d). In what follows we shall see that there are natural, necessary and sufficient conditions under which a feasible solution for problem (3) exists.

Observation 3: Problem (3) has a feasible solution if and only if

1) $c_l \geq \sum_{s:l\in L(s)} R_s$ for every link $l \in L$; and
2) $p_s D_s \geq \sum_{l\in L(s)} \left(\frac{(1-\beta)/\mu}{c_l} + \frac{\beta/\mu}{c_l - \sum_{s:l\in L(s)} R_s}\right)$ for every session $s \in S$.

Proof: Even though a rigorous algebraic proof is possible, we present here a short and more intuitive argument.

It is straightforward to verify that if these two conditions are satisfied, then a solution defined as $x_s = R_s$ for every session s, and $d_l = \frac{(1-\beta)/\mu}{c_l} + \frac{\beta/\mu}{c_l - \sum_{s:l\in L(s)} R_s}$ for every link l, satisfies (3a)-(3d).

It is readily seen that the first condition ensures that the link capacities are sufficient to support the corresponding link flows induced by allocating the minimum required rate to every session ($x_s = R_s$). Moreover, the second condition ensures that the end-to-end delay requirements (3b) is satisfied if every session is allocated its minimum required rate. Now assume that the first (respectively, the second) condition does not hold. This implies that the minimum rate allocation $x_s = R_s$ cannot be supported by the link capacities (respectively, cannot satisfy the delay requirements). Moreover, any other rate allocation that involves higher rates would need higher link capacities and would induce longer delays. Consequently, a rate allocation that satisfies (3a)-(3d) cannot exist. ■

It is worth noting that the conditions in Observation 3 are easy to compute, and can be used by the network to negotiate rate and delay requirements with the sessions prior to their admission. Once a set of sessions are admitted, their QoS requirements can be adhered to by the network because a feasible rate allocation to problem (3) is guaranteed to exist.

III. The Algorithm

In Section II, we have seen that the rate allocation problem as formulated by (3) is a convex optimization problem. Therefore, it can be solved in a centralized fashion by efficient interior-point methods [4]. In what follows, we describe a distributed rate allocation algorithm that is based on *Lagrangian decomposition* (λ_l denotes the price per unit bandwidth on link l).

Algorithm *Rate-Allocation*

- Initialize λ_l for every link l. Inmitialize the iteration count $k = 0$.
- Repeat the following three steps until convergence:

(1) $k := k + 1$. Every session s maximizes its individual benefit, by (selfishly) solving:

$$\max_{x_s \geq R_s} \quad U_s(x_s) - \left(\sum_{l\in L(s)} \lambda_l\right) x_s \tag{4}$$

(2) The network partitions the end-to-end delay requirements into per-link delays, by solving:

$$\begin{aligned} \text{minimize} \quad & \sum_l \frac{\lambda_l}{\mu d_l - \frac{1-\beta}{c_l}} \\ \text{subject to} \quad & \sum_{l\in L(s)} d_l \leq p_s D_s, \ \forall s && \text{(5a)} \\ & d_l \geq \frac{1}{\mu c_l} \ \forall l && \text{(5b)} \end{aligned}$$

(3) Every link l updates (independently) its price as follows:

$$\lambda_l := \left[\lambda_l - \alpha_k \big(c_l - \frac{\beta}{\mu d_l - \frac{1-\beta}{c_l}} - \sum_{s:l\in L(s)} x_s\big)\right]^+, \tag{6}$$

where α_k is the step size at iteration k ($\alpha_k = 1/\sqrt{k}$ is an appropriate choice).

In the following theorem we establish the convergence of algorithm *Rate-Allocation* to the global optimum of the overall rate allocation problem (3).

Theorem 1: Let $U_s(x_s)$ be strictly concave for every session s. Then algorithm *Rate-Allocation* converges always to the global optimal solution of problem (3).

Proof: We dualize the constraints given by (3a) to obtain the following Lagrangian relaxation of (3):

$$\begin{aligned} \max \quad & \sum_s U_s(x_s) + \sum_l \lambda_l \left(c_l - \frac{\beta}{\mu d_l - \frac{1-\beta}{c_l}} - \sum_{s:l\in L(s)} x_s\right) \\ \text{s.t.} \quad & \sum_{l\in L(s)} d_l \leq p_s D_s, \ \forall s && \text{(7a)} \\ & x_s \geq R_s, \ \forall s && \text{(7b)} \\ & d_l \geq \frac{1}{\mu c_l}, \ \forall l, && \text{(7c)} \end{aligned}$$

where λ_l are the dual variables. By rearranging terms in its objective function, it is readily seen that (7) decomposes into (4) and (5) for a given set of dual variables. Moreover, the dual variables in (7) are precisely the link prices used in (4) and (5). In other words, given a set of dual variables (link prices), solving (4) and (5) provides an optimal solution to (7).

Let $g(\lambda)$ denote the optimal objective function value of (7). The dual problem of (3) is, thus, given by

$$\min\ g(\lambda) \quad \text{s.t.} \quad \lambda_l \geq 0,\ \forall l. \tag{8}$$

The convexity of (3) implies that the problem has zero duality gap [4]. In other words, solving the dual problem (8) provides also an optimal solution to the primal problem (3).

It remains to show that the updates (6) solve the dual problem (8). In fact, it is not difficult to show that $c_l - \beta / \left(\mu d_l - \frac{1-\beta}{c_l} - \sum_{s:l\in L(s)} x_s \right)$ is a subgradient of $g(\lambda)$ with respect to λ_l. In other words, the updates (6) are subgradient updates [2] of the dual variables λ_l. By the convergence of subgradient algorithms for convex optimization problems [2], the iterations of algorithm *Rate-Allocation* will converge to an optimal solution to the dual (8). Because the primal (3) has zero duality gap and its objective function is *strictly concave*, the iterations of algorithm *Rate-Allocation* will also converge to the *unique* optimal solution to (3). ■

The following remarks are worth mentioning:

- To carry out Step (1), each source requires the knowledge its own utility function only.
- In Step (3), each link updates its price using local information only. Note that the aggregate flow crossing any link $\left(\sum_{s:l\in L(s)} x_s\right)$ can be measured locally.
- The network does *not* require the knowledge of the sessions' current data rates or utility functions to carry out Step (2).
- Consider the special case of $U_s(x_s) = \log(x_s)$ for every s. Inspecting the Karush-Kuhn-Tucker (KKT) conditions [4] leads then to the following closed-form solution for (4):

$$x_s = \max\left\{ R_s, \frac{1}{\sum_{l\in L(s)} \lambda_l} \right\}. \tag{9}$$

It is worth mentioning that the QoS rate allocation problem (3) is related to the basic network utility maximization (NUM) problem studied by Kelly *et. al.* in [12] and [13]. In particular, (3) reduces to the basic NUM problem if all QoS constraints are removed. Moreover, algorithm *Rate-Allocation* and its derivation (in the proof of Theorem 1) come as an application to standard algorithm (see, e.g., [17]) to solve the basic NUM problem. The iterations of the basic NUM algorithm in [17]) can be broadly stated as follows:

1) *Session algorithm:* every session maximizes its own benefit by solving a problem similar to (4).
2) *Link algorithm:* every link updates its price per unit bandwidth using a formula similar to (6).

Low *et. al.* [16], [18] showed that existing TCP and (active) queue management (AQM) protocols can be interpreted as distributed algorithms that implicitly solve the basic NUM problem and its dual. In particular, the *session algorithm* is carried out by TCP and the *link algorithm* is carried out by an AQM scheme.

Similarly, algorithm *Rate-Allocation* can be viewed as a generalization of TCP/AQM algorithms to handle end-to-end QoS. In particular, Step (1) can be viewed as a *TCP* algorithm that adapts session data rates in response to congestion information that is provided to the sessions in form of link prices. Notice that if some session s has $\log(x_s)$ as its utility function[1], then the corresponding TCP algorithm of Step (1) would be simply to adapt the rate of session s according to (9). Step (3) can be viewed as an *AQM* algorithm that, similarly to REM [1], updates the price of each link with the purpose of providing the session sources with a link congestion measure. Step (2), in which the network partitions the end-to-end delay requirements into local link delays, is the additional component needed to handle end-to-end QoS.

IV. Deterministic Delay Requirements and PGPS

Now, we consider the QoS rate allocation problem when nodes employ PGPS [20] schedulers. In particular, we assume that every link $l \in L$ is served by a PGPS scheduler at the source node of the link. Therefore, sessions are associated with service weights at each link, and receive service in proportion to their respective weights. As a result, every session s is guaranteed a certain data rate on every link along its route. Under PGPS scheduling, sessions need not to be allocated the same rate on every link along their respective routes.

We also assume that the traffic of every session s is shaped by a token bucket [3] shaper with parameters (b_s, R_s). In other words, the maximum burst size of session s is b_s (bits) and its long term average data rate is R_s (bits/s).

Under PGPS scheduling, the worst-case end-to-end delay T_s of session s can be bounded by [10], [23]:

$$T_s \leq \frac{b_s}{\min_{l\in L(s)}\{x_s^l\}} + \sum_{l\in L'(s)} \frac{M_s}{x_s^l} + \delta_s, \tag{10}$$

where

x_s^l	rate allocated to session s on link $l \in L(s)$ (in bits/s);
$L'(s)$	partial route used by session s that consists of all links in $L(s)$ except the last hop;
M_s	maximum packet length of session s (in bits);
δ_s	constant that is completely characterized by the scheduling algorithm and the propagation delay along the route of session s.

In particular, $\delta_s = \sum_{l\in L(s)} \left(\frac{\max_{s'\in L(s)}\{M_{s'}\}}{c_l} + \tau_l \right)$, where τ_l is the propagation delay on link l.

Let D_s denote the worst-case delay requirement of session s, and let U_s be the utility of session s as a function of the smallest (bottle-neck) data rate allocated to s. We express the

[1]It has been shown that sessions that use TCP Vegas have an implicit utility function that is proportional to $\log(x_s)$.

QoS rate allocation problem as follows:

$$\begin{aligned} \max \quad & \sum_s U_s(x_s) \\ \text{s.t.} \quad & \sum_{s:l\in L(s)} x_s^l \le c_l,\ \forall l & \text{(11a)} \\ & \frac{b_s}{x_s} + \sum_{l\in L'(s)} \frac{M_s}{x_s^l} + \delta_s \le D_s,\ \forall s & \text{(11b)} \\ & x_s^l \ge R_s,\ \forall s\in S, l\in L(s) & \text{(11c)} \\ & x_s \le x_s^l,\ \forall s\in S, l\in L(s), & \text{(11d)} \end{aligned}$$

where the *variables* are:

x_s^l rate allocated to session s on link $l \in L(s)$ (in bits/s);

x_s smallest data rate allocated to session s, i.e., $x_s = \min_{l\in L(s)} x_s^l$.

The objective in (11) is to maximize the total network utility. Constraint set (11a) ensures that the total flow on each link is less than its capacity. Note that it is also implicitly assumed that $\sum_{l\in L(s)} R_s < c_l$. Therefore, (11a) ensures also that network of PGPS servers is *globally stable*. Constraints (11b) represent the end-to-end delay constraints. Constraint set (11c) ensures that the minimum rate requirements are met. Each session is, thus, ensured to be *locally* stable. Constraint set (11d) implies that, at the optimal solution, $x_s = \min_{l\in L(s)}\{x_s^l\}$, i.e., x_s is precisely the rate allocated to session s on its bottleneck link. In particular, assume that the optimal solution of (11) satisfies $x_s < \min_{l\in L(s)}\{x_s^l\}$. Consequently, the objective function value can be increased by increasing x_s to $\min_{l\in L(s)}\{x_s^l\}$ without violating any constraint.

Note that admission control techniques, e.g., the technique in [6], have to be used to decide wether (11) has a feasible solution or not.

Now we make the following observation on problem (11). The proof is omitted due to space limitations.

Observation 4: Problem (11) is a convex optimization problem as long as the utility functions $U_s(x_s)$ are convcave.

Convexity of (11) implies that the problem has zero duality gap [2]. This gives rise to the use of a Lagrangian decomposition approach to solve (11). In particular, the following algorithm will converge to the *unique* optimal solution of (11) as long as the utilities $U_s(x_s)$ are *strictly concave* (the proof is almost identical to Theorem 1).

Algorithm *Rate-Allocation-PGPS*

- Initialize λ_l for every link l. Initialize the iteration count $k = 0$.
- Repeat the following two steps until convergence:

(1) $k := k + 1$. Every session s selfishly solves:

$$\begin{aligned} \max \quad & U_s(x_s) - \sum_{l\in L(s)} \lambda_l x_s^l \\ \text{s.t.} \quad & \frac{b_s}{x_s} + \sum_{l\in L'(s)} \frac{M_s}{x_s^l} + \delta_s \le D_s & \text{(12a)} \\ & x_s^l \ge R_s,\ \forall l\in L(s) & \text{(12b)} \\ & x_s \le x_s^l,\ \forall l \in L(s). & \text{(12c)} \end{aligned}$$

(2) Every link l updates (independently) its price as follows:

$$\lambda_l := \left[\lambda_l - \alpha_k(c_l - \sum_{s:l\in L(s)} x_s^l)\right]^+, \qquad (13)$$

where α_k is the step size at iteration k ($\alpha_k = 1/\sqrt{k}$ is an appropriate choice).

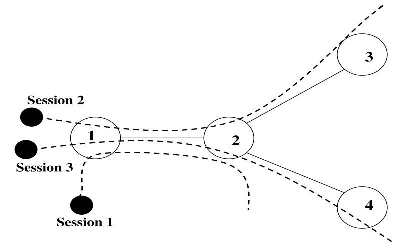

Fig. 1. A simple network with 4 nodes, 3 links and 3 sessions.

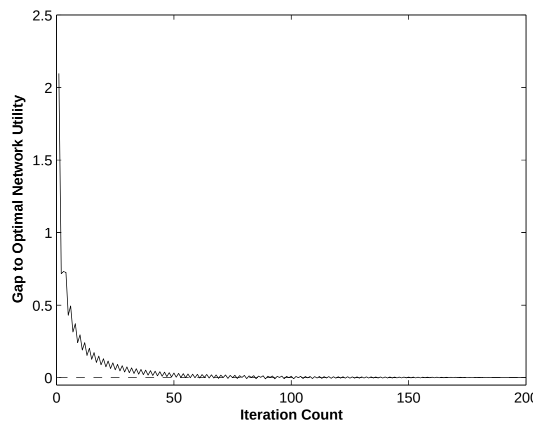

Fig. 2. Convergence of the total network utility.

Again, algorithm *Rate-Allocation-PGPS* can be considered as a generalization of TCP/AQM algorithms (as interpreted by Low [16]) to handle end-to-end QoS. In particular, the rate update in Step (1) and the link price update in Step (2) can be viewed as generalizations to TCP and AQM algorithms, respectively.

V. Numerical Results

The objective of this numerical study is to illustrate the convergence of algorithm *Rate-Allocation*.

We use a simple network with 4 nodes, 3 links and 3 (unicast) sessions as shown in Fig. 1. Session 1 uses path 1-2, Session 2 uses path 1-2-3 and session 3 uses path 1-2-4. Without loss of generality, we assume that the capacity of each link is 1 Mbits/s, every session has a minimum data rate requirement of 250 Kbits/s, and every session has an *average* delay requirement of 0.0336 seconds (per packet). We also assume that the utility function for session s (=1,2,3) is given by $U_s(x_s) = \log(x_s)$.

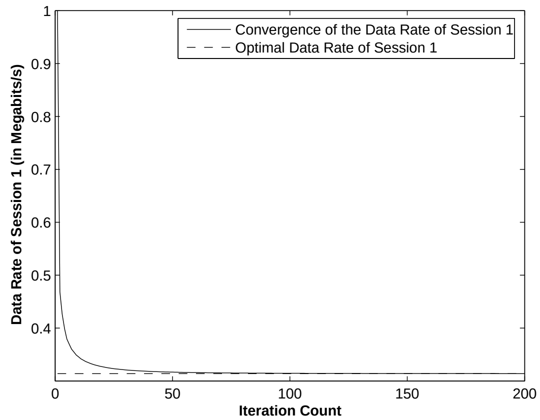

Fig. 3. Convergence of the data rate of Session 1.

We assume a general packet length distribution with mean 420 bytes and standard deviation 521 bytes[2]. In this case, $\beta = 0.5$. Fig. 4 depicts the difference between the total network utility obtained at each iteration of algorithm *Rate-Allocation* and the optimal total network utility obtained by solving (3). In fact, after 200 iterations the total network utility is within 0.021% of the optimal value.

Moreover, after 200 iterations, the data rates of Session 1, Session 2 and Session 3 are within 0.02%, 0.019% and 0.019% of their respective optimal values. As an example, Fig. 5 illustrates the convergence of the data rate of Session 1 to the optimal value of 314 Kbits/s.

VI. Conclusion

We addressed the problem of allocating data rates to a set of network sessions with end-to-end bandwidth and delay requirements. We gave a unified convex programming formulation that captures both *average* and *probabilistic* delay requirements. Moreover, we presented a distributed algorithm and established its convergence to the global optimum of the overall rate allocation problem. In our algorithm, *session sources* update their rates as to maximize their individual benefit (utility minus bandwidth cost), the *network* partitions end-to-end delay requirements into local per-link delays, and the *links* adjust their prices to coordinate the sources' and network's decisions, respectively. Our algorithms came as an application of a known Lagrangian decomposition approach for solving the basic network utility maximization problem [12], [13]. The algorithm can thus be viewed as a generalization of TCP and queue management algorithms (as interpreted by Low [16]) to handle end-to-end QoS. We also extended our results to deterministic delay requirements when nodes employ PGPS schedulers.

[2]These values reflect the length distribution of Internet packets seen at NASA Ames Internet Exchange (AIX) in February 2000. See `http://www.caida.org/analysis/AIX/plen_hist/`.

References

[1] S. Athuraliya, S. Low, V.H. Li and Q. Yin, "REM: Active Queue Management", *IEEE Network*, vol. 15, no. 3, pp. 48–53, May/June 2001.
[2] D. Bertsekas, *Nonlinear programming*, Athena Scientific, 1999.
[3] D. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hal, Inc, 1992.
[4] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
[5] K.M.F. Elsayed, "A Framework for End-to-End Deterministic-Delay Service Provisioning in Multiservice Packet Networks", *IEEE Transactions on Multimedia*, vol. 7, no. 3, pp. 563–571, June 2005.
[6] K.M.F. Elsayed, A. Saad and M.T. El-Hadidi, "Performance Evaluation of Resource Reservation and Call Admission Policies for Deterministic Services in PGPS-Based Packet Networks ", *Computer Communications*, vol. 7, no. 3, pp. 1513–1526 , Nov. 2002.
[7] V. Firoiu and D. Towsley, "Call Admission Control and Resource Reservation for Multicast Sessions", in *Proc. IEEE INFOCOM*, pp. 94–101 , March 1996.
[8] M. Gerla and L. Kleinrock, "On the Topological Design of Ditributed Computer Networks", *IEEE Transactions on Communications*, vol. COM-25, no. 1, pp. 48–60, Jan. 1977.
[9] K. Gopalan, T-c. Chiueh and Y-J. Lin, "Delay Budget Partitioning to Maximize Network Resource Usage Efficiency", in *Proc. IEEE INFOCOM*, pp. 2060–2071 , March 2004.
[10] P. Goyal and R.M. Vin, "Generalized Guaranteed Rate Scheduling Algorithms: A Framework", *IEEE/ACM Transactions on Networking*, vol. 5, no. 4, pp. 561–571, Aug. 1997.
[11] N. Jin, G. Venkitachalam and S. Jordan, "Dynamic Congestion-Based Pricing of bandwidth and Buffer", *IEEE/ACM Transactions on Networking*, vol. 13, no. 6, pp. 1233–1246, Dec. 2005.
[12] F.P. Kelly, "Charging and Rate Control for Elastic Traffic", *European Transactions on Telecommunications*, vol. 8, pp. 33–37, 1997.
[13] F.P. Kelly, A. Maulloo and D. Tan, "Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability", *Journal of the Operations Research Society*, vol. 49, no. 3, pp. 237–252, March 1998.
[14] N.J. Keon and G. Anandalingam, "Optimal Pricing for Multiple Services in Telecommunications Networks offering Quality-of-Service Guarantees", *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 66–80, Feb. 2003.
[15] D.H. Lorenz and A. Orda, "Optimal Partitions of QoS Requirements on Unicast paths and Multicast Trees", *IEEE/ACM Transactions on Networking*, vol. 10, no. 1, pp. 102–114, Feb. 2002.
[16] S.H. Low, "A Duality Model of TCP and Queue Management Algorithms", *IEEE/ACM Transactions on Networking*, vol. 11, no. 4, pp. 525–536, Aug. 2003.
[17] S.H. Low and D.E. Lapsley, "Optimization Flow Control, I: Basic Algorithm and Convergence", *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 861–874, Dec. 1999.
[18] S.H. Low, L.L. Peterson and L. Wang, "Understanding Vegas: A Duality Model", *Journal of the ACM*, vol. 49, no. 2, pp. 2072–235, March 2002.
[19] L. Massoulié and J. Roberts, "Bandwidth Sharing: Objectives and Algorithms", *IEEE/ACM Transactions on Networking*, vol. 10, no. 3, pp. 320–328, June 2002.
[20] A.K. Parekh and R.G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: the Multiple Node Case", *IEEE/ACM Transactions on Networking*, vol. 2, no. 2, pp. 137–150, April 1994.
[21] S. Sarkar and L. Tassiulas, "Fair Distributed Congestion Control in Multirate Multicast Networks ", *IEEE/ACM Transactions on Networking*, vol. 13, no. 1, pp. 121–133, Feb. 2005.
[22] S. Siachalou and L. Georgiadis, "Algorithms for Precomputing Widest Paths and Multicast Trees", *IEEE/ACM Transactions on Networking*, vol. 13, no. 5, pp. 1174–1187, Oct. 2005.
[23] D. Stiliadis and A. Varma, "Latency-Rate Servers: A General Model for Analysis of Traffic Scheduling Algorithms ", *IEEE/ACM Transactions on Networking*, vol. 6, no. 5, pp. 611–624, Oct. 1998.
[24] S. Verma, R. K. Pankaj and A. Leon-Garcia, "Call Admission and Resource Reservation for Guaranteed Quality of Service (GQoS) Services in Internet", *Computer Communications*, vol. 21, no. 4, pp. 362–374, April 1998.