Differential Join Prices for Parallel Queues: Social Optimality, Dynamic Pricing Algorithms and Application to Internet Pricing

Parijat Dube
Project MISTRAL INRIA, 2004
Route des Lucioles, 06902, B.P. 93
Sophia Antipolis Cedex FRANCE
Parijat.Dube@sophia.inria.fr

Vivek S. Borkar
School of Technology & Computer Science
Tata Institute of Fundamental Research
Homi Bhabha Road Mumbai-400 005 INDIA
borkar@tifr.res.in

D. Manjunath
Deptt. of Electrical Engineering
Indian Institute of Technology,
Powai Mumbai- 400 076, INDIA
dmanju@ee.iitb.ernet.in

***Abstract*— We consider a system of identical parallel queues served by a single server and distinguished only by the price charged at entry. A Poisson stream of customers joins the queue by a greedy policy that minimizes a 'disutility' that combines price and congestion. A special case of linear disutility is analyzed for which it is shown that the individually optimal greedy queue join policy is nearly socially optimal. For this queueing system, a Markov decision theoretic framework is formulated for dynamic pricing in the general case. This queueing system has application in the pricing of Internet services.**

I. INTRODUCTION

Traditional pricing models for Internet services have been fairly simple such as flat rate or connect time charges for the retail user and a flat rate depending on the interconnection bandwidth for the bulk user. These pricing models do not account for the traffic volume or the distances traveled by these traffic as is done, for example, in the telephone network or provide differential grades of service in terms of the bandwidth, delay or packet drops. It is now well recognized that the traditional pricing structures of the Internet are not economically sustainable because the volume and variety of traffic on the Internet is growing at an explosive rate and the issue of pricing is becoming increasingly important. This is not only because of the economic imperatives, but also because pricing promises to provide a relatively simple mechanism for traffic control in a system that is far too complex to be amenable to the analytic tools of traditional control theory. Thus, not surprisingly, there is a rapidly increasing body of literature on pricing of the Internet. However, it is also true that the Internet traffic process is less amenable to sophisticated pricing models. For example, of the four categories of charges – access, usage, congestion and quality (see [16]) it would be computationally expensive to develop pricing structures for all but the access charges. Albeit, there has been some effort at developing mechanisms for usage based billing. Currence et al discuss the pros and cons of usage based billing for IP networks and also review some commercial usage based billing products [6]. There is also a large body of literature on congestion charges. For example, Kelly considers a congestion price model where different users get a share of the rate depending on the prices that they are willing to pay [10]. The share also depends on the maximization of a utility function. In this paper we consider a model for pricing of quality. Specifically, we consider pricing in a multiclass service system like the diffserv model of the IETF [3], and our pricing scheme is compatible with it.

In the differential service model, the basic idea is to divide the available bandwidth among the multiple classes of traffic in a static or dynamic manner. Each node in the network maintains a separate logical queue for each class for each outgoing link and services them according to the rules of the bandwidth sharing policy. The simplest way to provide multiclass service is through the traditional priority queue model. A lower class queue is not served until the queues of the higher classes are empty. We could have preemptive and non preemptive service. This has the obvious disadvantage that the lower priority queues can be starved for extended periods of time. An alternative to the priority model is proposed by Odlyzko [13] which is more friendly to the lower priority customers and is called the Paris Metro Pricing scheme (PMP). In the PMP scheme the network is logically partitioned in a static manner with each partition being allocated a fixed share of the network resources and the access price for each class is differentially priced. The specific case of equal division of resources among the multiple classes is similar to that used by the Paris Metro till recently when the first class was abolished. A similar model is proposed in [2] and is called the Olympic pricing scheme. An excellent overview and bibliography on Internet pricing issues and models can be found in [9].

Our aim here is to propose a very simple scheme that is appealing in its simplicity of implementation and provide some theoretical justification for it. We cannot, however, claim any originality for the scheme, as it is fashioned after a queue management scheme already in practice at Tirupati, one of the major pilgrimage centers in India. Suffice to say that it has been operating there with an efficiency that leaves even a casual visitor greatly impressed. With this background in mind, we dub it the 'Tirupati scheme'. In the following we discuss the scheme with reference to a queueing system but is easy to see that our discussions includes the case of a feedforward (i.e., loop-free) network.

The contribution of this paper is to first analyze the social optimality of the proposed Tirupati pricing model. We will see that the user behaviour in the Tirupati pricing model is similar to that in a system of identical parallel queues for which it is well known that the "join shortest queue" join policy is both individually and socially optimal. We will show that the difference between the social cost of the optimally priced system and that of the Tirupati system is $\bar{K}\epsilon$ for constants $\bar{K}$ and ϵ that will be defined later. Another, possibly more practical, contribution of this paper is the dynamic pricing using a dynamic programming equation and a reinforcement learning based online pricing algorithm.

The paper is organized as follows. In the next section we describe the general Tirupati queue pricing model and specialize it to an analytically amenable 'linear' model for a single class of customers. In Section III we analyze this simplified model. Some extensions are also pointed out. Section IV introduces the dynamic pricing problem in the general framework. Section V introduces learning algorithm for dynamic pricing and Section VI provides some simulation results on its performance. We conclude in Section VII.

II. THE TIRUPATI PRICING MODEL

The simplest formulation of the Tirupati scheme is as follows: We have a single server serving $K > 1$ queues with a total service rate $\bar{\mu}$. The service rate is equally shared among the non empty queues. The K queues have different join prices $p_1 \leq p_2 \leq \cdots \leq p_K$. The arrivals form a Poisson process according to a rate λ. The customers belong to $M \geq 1$ distinct types, with the type of an arriving customer being independent of other customers. On arrival, a customer is informed of the queue length and prices of the different queues and joins the queue that offers the least *disutility* (to be defined later), any tie being resolved by randomization. The queues are served in a round robin fashion (non preemptive), with FCFS discipline within each queue.

The Tirupati scheme is similar in spirit to the multiclass pricing schemes like the Paris Metro scheme [13] or the Olympic pricing scheme [2] except for the following differences. Firstly, it is easy to see that the Tirupati scheme as defined above is work conserving. This is different from the Paris Metro Pricing model where the capacity of the server is partitioned among the classes. Also, in the Tirupati scheme the congestion level in terms of the queue length of each of the queues is posted to enable the customers to calculate their disutility. Also, the above two schemes do not explicitly involve a model of customer behaviour based upon a "disutility" calculated from the posted prices and congestion levels. Further, note that it retains the spirit of the Internet in being simple to implement with a flat pricing model similar to the PMP.

The extension of this queueing system to the Internet is fairly straightforward and we will continue our discussion on the queueing system. Our analysis is based on Markov decision theory. See [5] for an earlier work in this spirit.

A. Model for Analysis

For the purpose of analysis we make a simplification of the general Tirupati model by replacing the single queue by K parallel queues each with an effective service rate of $\mu = \bar{\mu}/K$. What this ignores is the fact that when one or more queue is empty, the extra service made available would get evenly distributed over the remaining queues in round robin, not so in the simplified picture above. We opt for the simplification because it simplifies the analysis and also because in the heavy traffic situations one is interested in, empty queues would be sufficiently rare so as to cause small error. We make this precise later. (See comments at the end of Section III.)

Consider a customer of type m arriving at time t. Let $Q_i(t)$ be the number of customers in queue i, $1 \leq i \leq K$ and let $\mathbf{Q}(t) = [Q_1(t), Q_2(t), \ldots, Q_K(t)]^T$ be the overall queue length vector. $\mathbf{Q}(t^-)$ will denote the queue vector 'just before t'. The disutility of the i-th queue for the above customer is given by $\Psi_i(Q_i(t), p_i, m)$, where the map Ψ is increasing in the first two arguments. We also admit

- *Balking*: Constants $\alpha_{im} > 0$ can be prescribed such that a type m customer balks from joining queue i if its disutility for that queue is strictly greater than α_{im}. If this is true for all i, the customer leaves without joining any queue, or simply '*balks*'.
- *Finite buffers*: Queue i can have finite buffers to allow at most B_i customers at any time.

Thus the above customer joins the i-th queue if i minimizes $\Psi_j(Q_j(t^-), p_j, m)$ among all j for which $\Psi_j(Q_j(t^-), p_j, m) \leq \alpha_{jm}$ and $Q_j(t^-) < B_j$. We call this the 'Join the Minimum Cost Queue' policy, JMCQ for short. This policy is clearly optimal for this customer alone, regardless of what the customers before or after do. That is, it is *individually optimal*. Note also that the balking mechanism can either be customer induced (i.e., reflecting price/congestion sensitivity of the customer) or system induced (i.e., reflecting implicit prioritization by the system).

We now specialize to the case when $m = 1$, i.e., a single type, and the disutility is linear: $\Psi_i(q, p_i, 1) = q_i + p_i$. For notational simplicity, we denote it as $\Psi_i(q)$. We further assume that the p_is are integers. The customer does not join queue i whenever the disutility strictly exceeds $\beta_i = \min(B_i + p_i - 1, \alpha_{i1})$. In the latter case, we shall say that queue i is 'blocked'. This case is amenable to analysis, which is provided in the next section. The aim is to evaluate the performance of JMCQ in comparison with other queue join policies. For this purpose, the most general class of policies we consider will be the nonanticipative policies, represented by a process $\{Z_n\}$ taking values in $\{0, 1, 2, \ldots, K\}$, with the interpretation: Z_n is the index of the queue joined by the n-th arrival. Z_n is set equal to 0 if the $n-$th customer balks. We shall call the queue join policy *Markov* if $Z_n = v(\mathbf{Q}(\tau_n))$ for a suitable map $v(\cdot)$, where τ_n is the arrival instant of the $n-$th customer. Note that the JMCQ that we discuss above and the 'Join the Shortest Queue' (JSQ for short) are nonanticipative Markov policies. The JSQ policy without queue join prices has been analyzed in literature. See, for example [7] and [12].

Remark 1: Even for a single class of customers, offering differentiated services may make sense because it gives the 'system manager' flexibility in optimizing revenue without significant loss of customers. We can argue that a simple monotone increase of waiting times in a single queue will lead to increased balking.

For any queue join process Z_n, we introduce the *social cost*,

$J(\{Z_n\})$ defined as follows

$$J(\{Z_n\}) = \limsup_{n\to\infty} \frac{1}{n} \sum_{m=1}^{n} E[\Psi_{Z_n}(\mathbf{Q}(\tau_m))].$$

From the above definition, the social cost is essentially the average disutility of the customers in the limit. In case of a Markov policy $Z_n = v(\mathbf{Q}(\tau_n)), n \geq 1$, $\mathbf{Q}(\cdot)$ is a time-homogeneous Markov chain with a unique stationary distribution and by PASTA (Poisson Arrivals See Time Averages), the above will equal $E[\Psi_{v(\mathbf{Q}(\tau_n))}(\mathbf{Q}(\tau_n))]$ where the expectation is with respect to the stationary distribution. For JMCQ, this reduces to

$$E[\min_i \Psi_i(\mathbf{Q}(\tau_n))] = E[\min_i (Q_i(\tau_n) + p_i)].$$

Letting $\{t_n\}$ denote the successive event times (i.e., arrival or potential departure times - a potential departure is a true departure if and only if the queue in question is nonempty), we can consider the problem of controlling the Markov decision process $\{\mathbf{Q}(t_n)\}$ so as to minimize the social cost. Standard Markov decision theoretic results [14] then tell us that it suffices to consider Markov policies. Our main result, proved in the next section, will be that *under heavy traffic assumption, JMCQ is nearly optimal*.

III. Analysis of the Linear Case

The analysis will be based upon a comparison with another process which we call the pseudo-queue. This process, denoted by $\tilde{\mathbf{Q}}(\cdot) = [\tilde{Q}_1(\cdot), \cdots, \tilde{Q}_K(\cdot)]$, has the same dynamics as the original queue, but $\tilde{Q}_i(t)$ is allowed to take values $\{j : j \geq -p_i\}$. That is, we allow departures out of an empty queue so that the queue length can go negative, as long as it does not drop below the negative price. Let $\hat{\mathbf{Q}}(\cdot) = [\hat{Q}_1(\cdot), \cdots, \hat{Q}_K(\cdot)]$ be defined by $\hat{Q}_i(t) = \tilde{Q}_i(t) + p_i, 1 \leq i \leq K$. Then $\hat{\mathbf{Q}}(\cdot)$ is a legal (i.e., nonnegative) queue length process and JMCQ policy for $\tilde{\mathbf{Q}}(\cdot)$ corresponds to the JSQ policy for $\hat{\mathbf{Q}}(\cdot)$.

Let $S = \Pi_{i=1}^{K}\{0, 1, \cdots, B_i\}$, $\tilde{S} = \Pi_{i=1}^{K}\{-p_i, -p_i + 1, \cdots, B_i\}$ and $\hat{S} = \Pi_{i=1}^{K}\{0, 1, \cdots, B_i + p_i\}$ denote the respective state spaces for $\mathbf{Q}(\cdot)$, $\tilde{\mathbf{Q}}(\cdot)$ and $\hat{\mathbf{Q}}(\cdot)$.

Define

$$\tilde{J}(\{Z_n\}) = \limsup_{n\to\infty} \frac{1}{t} \int_0^t E\left[\sum_{i=1}^{K} (\tilde{Q}_i(s) + p_i)\right] ds$$

and

$$\hat{J}(\{Z_n\}) = \limsup_{n\to\infty} \frac{1}{t} \int_0^t E\left[\sum_{i=1}^{K} \hat{Q}_i(s)\right] ds.$$

Lemma 1: 1) JMCQ minimizes $\tilde{J}(\{Z_n\})$
2) JSQ minimizes $\hat{J}(\{Z_n\})$

Given the relationship between the pseudo-queue $\tilde{\mathbf{Q}}(\cdot)$ and the queue $\hat{\mathbf{Q}}(\cdot)$, the two claims are clearly equivalent and it suffices to prove either. We prove part 2 in the Appendix by adapting the 'forward induction' argument of [15], section 8.3.

Corollary 1: For the pseudo-queue, JMCQ minimizes the social cost.

Proof: By standard Markov decision theory, it suffices to consider the minimization over Markov policies. By the relationship between $\tilde{\mathbf{Q}}(\cdot)$ and $\hat{\mathbf{Q}}(\cdot)$, it suffices to show that JSQ minimizes $E[\hat{\mathbf{Q}}_{Z_n}(\tau_n)]$ over all Markov policies, where the expectation is w.r.t. the corresponding stationary distribution. Recall that Z_n is the index of the queue which the n-th customer joins. By Little's theorem, under a Markov policy,the following relation holds between the corresponding stationary expectations:

$$E\left[\sum_{i=1}^{K} \hat{Q}_i(\tau_n)\right] = \lambda \left(\frac{K}{\mu} E\left[\hat{Q}_{Z_n}(\tau_n) + 1\right]\right),$$

where the term in the parentheses is the mean waiting time.

The claim now follows from Lemma 1. □

To compare the social costs of the real queue and the pseudo-queue, we shall first compare their stationary distributions under a common Markov policy and heavy traffic conditions. The latter refers to :

Heavy Traffic Assumption: There exists a 'small' $\epsilon > 0$ such that under any Markov policy, the stationary probability of the event $\{Q_i(t) = 0 \text{ for some } i\}$ is less than ϵ.

This requires that λ should not be much smaller that μ.

Lemma 2: Under the heavy traffic conditions, there exists a constant $\bar{K} > 0$ such that under any Markov policy, the social costs for the real queue and the pseudo-queue do not differ by more than $\bar{K}\epsilon$.

Proof: Fix a Markov policy. Furthermore, using the fact that $S \subset \tilde{S}$, view $\mathbf{Q}(\cdot)$ as an $\tilde{S}-$valued chain, with transition probabilities out of states in $\tilde{S} - S$ being the same as those for the pseudo-queue. Note that these additional states will be transient for $\mathbf{Q}(\cdot)$ and thus have zero probability under the stationary distribution. Let $P, \hat{P}$ denote the transition probability matrices for $\mathbf{Q}(\cdot), \tilde{\mathbf{Q}}(\cdot)$ respectively and $\pi, \hat{\pi}$ the respective stationary distributions, written as column vectors. Letting e denote the $|\tilde{S}|-$dimensional vector of all ones, we then have

$$\begin{aligned} (I - P^T + ee^T)\pi &= e, \\ (I - \hat{P}^T + ee^T)\hat{\pi} &= e. \end{aligned}$$

Thus

$$\pi - \hat{\pi} = (I - \hat{P}^T + ee^T)^{-1}(P^T - \hat{P}^T)\pi,$$

from which it follows that $||\pi - \hat{\pi}|| < \bar{K}\epsilon$ for a suitable $\bar{K} > 0$. We use the fact that P and $\hat{P}$ differ only on rows corresponding to states that have at least one queue empty and $P - \hat{P}$ is non zero only in those rows and zero in other rows. The claim follows. □

Corollary 2: The minimum of the social cost for $\mathbf{Q}(\cdot)$ and $\tilde{\mathbf{Q}}(\cdot)$ respectively differ by at most $\bar{K}\epsilon$ for $\bar{K}$ and ϵ as above.

Proof: Immediate from above. □

Theorem 1: The social cost of JMCQ for $\mathbf{Q}(\cdot)$ is within $2\bar{K}\epsilon$ of the optimum for $\bar{K}, \epsilon$ as above.

Proof: This follows on combining Corollary 1 with Lemma 2 and Corollary 2 □

This result states that JMCQ is nearly optimal under heavy traffic conditions for our simplified version of the Tirupati pricing model.

We conclude this section by pointing out some immediate extensions.

1) The approximation already made in the beginning of the article, viz., passing from round robin between nonempty queues *a la* Tirupati to K parallel queues, also introduces an error that is $O(\epsilon)$. This can be shown along lines similar to the above. Thus the above in fact extends to the case when the available service is split among nonempty queues in a round robin fashion.
2) The arguments above and in the Appendix extend to the 'processor sharing model', more relevant for virtual circuit switched networks where the queues of the above model may correspond to virtual paths (VPs) and the customers correspond to virtual circuits (VCs). The link bandwidth is divided among the VPs and the bandwidth available to each VC will be the bandwidth available to the VP evenly divided among all the VCs in the VP. This is because the analysis depends only on the arrival and departure processes, which remain the same. Also, the rationale for the above choice of disutility remains the same.

Remark 2: It would be interesting to extend Theorem 1 to the multiclass case. One particular case of multiclass is when the threshold $\alpha_{i,m} = \alpha_m \ \forall i$ for class m. The social optimality of JMCQ extends to this case of multiclass customers and can be proved along the lines of proof for Theorem 1.

IV. Dynamic Pricing

Dynamic pricing refers to the case when the 'system manager' adjusts the price vector with time to maximize revenue. It is reasonable to assume that customers follow the individually optimal JMCQ policy, which fixes their behaviour *given* the pricing policy. Thus from the point of view of the system manager, it is a Markov decision process, described in detail below. We shall consider the general set-up introduced in the beginning of the paper, not just the single type, linear case.

Standard Markov decision theoretic arguments show that we may consider the embedded discrete time Markov chain $\{\mathbf{Q}(t_n)\}$ controlled by the process of dynamically adjusted prices $\{p^n = [p_1^n, \cdots, p_K^n]\}$, where for each n, $p_i^n \in \mathbb{P}_i$ and is the price for joining queue i at the nth event time. At each event time t_n, there is a potential departure from the $i-$th queue with probability $\frac{\mu}{(\lambda+\mu)K}$, which is a real departure if the queue is nonempty. With probability $\frac{r_m\lambda}{\lambda+\mu}$ there is an arrival of type m, who joins the queue j that minimizes $\Psi_j(Q_j(t_n^-), p_j^n, m)$ among those queues which are not blocked for his type, if any, and balks otherwise. For simplicity, we shall assume that $\mathbb{P}_i^m$ is compact. The systems manager's aim then is to maximize the expectation of the revenue

$$\limsup_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} E\left[c(Z_n, \mathbf{Q}(t_n^-), \mathbf{Q}(t_n))\right],$$

where $c(Z_n, \mathbf{Q}(t_n^-), \mathbf{Q}(t_n))$ is the price paid by an arrival at n-th event time if this is an arrival instant and zero otherwise. Z_n is a K valued "control" process (the price vector at the n-th event time).

Observe that $\{\mathbf{Q}(t_n^-)\}$ is an irreducible Markov chain under every stationary policy. This is a standard 'average cost' Markov decision problem [14] with the associated dynamic programming equations given below. Let e_k for $1 \le k \le K$ denote the $K-$dimensional unit vector in the k-th coordinate direction. The dynamic programming equations are a

$$\begin{aligned} V(q) + \gamma \;\; &= \;\; \max_{p\in U} \frac{\lambda}{\lambda+\mu} \sum_{m=1}^{M} r_m \\ & \sum_{k=1}^{K} \frac{1}{|\text{Argmin}_j \Psi_j(q_j, p_j, m)|} (p_k + V(q+e_k)) \times \\ & \times \mathbf{I}\{\Psi_k(q_k, p_k, m) \le \min(\Psi_j(q_j, p_j, m), \\ & j \ne k, B_k + p_k - 1, \alpha_{km})\})] + \\ & \frac{\mu}{(\lambda+\mu)K} \sum_{k=1}^{K} V(q - e_k \mathbf{I}\{q_k > 0\}), \end{aligned}$$

where γ is the optimal revenue. It is known that these equations specify γ uniquely and $V : S \to \mathcal{R}$ uniquely up to an additive constant. Also, if $p = \hat{p}(q)$ minimizes the r.h.s., then $Z_n = \hat{p}(Q(t_n))$ is the optimal pricing policy.

Thus the optimal policy is known if one can compute $V(\cdot)$. This can be done by standard methods such as value iteration, policy iteration or linear programming described in [14]. In case the transition probabilities are not known, one can employ simulation-based approximate methods based on reinforcement learning, see, e.g., [1], [11]. This is an important situation, because transition probabilities depend not only on the arrival process and the service rates, but also on the disutility functions which may not be known even approximately. In the next section we present an on-line algorithm for solving the dynamic programming equation (1). The algorithm belongs to the class of actor-critic algorithms proposed in [8]. It may be noted that although we propose this as an online adaptive algorithm, it can also be used for offline learning based on simulation using accumulated statistics.

V. An Actor-Critic Type Algorithm

The state space $\mathbb{Q}$ of our MDP is finite. We have a finite action space $\mathbb{P}$, which is the possible set of values for the price vectors. Let $\rho_{xy}(\overline{p})$ denote the probability that the next state is y, given that the current state is x and the current action (the value of price vectors) is $\overline{p} \in \mathbb{P}$. A *randomized stationary policy* (RSP) is a mapping μ that assigns to each state x a probability distribution over the action space $\mathbb{P}$. We consider a set of randomized stationary policies $\{\mu_\theta; \theta \in \mathbb{R}^n\}$, parametrized in terms of a vector θ. For each pair $(x, u) \in \mathbb{X} \times \mathbb{P}$, $\mu_\theta(x, \overline{p})$ denotes the probability of taking action $\overline{p}$ when the state x is encountered, under the policy corresponding to θ. Under the assumptions (see ASSUMPTION 2.1 in [8]) for each $\theta \in \mathbb{R}^n$ consider the average cost function $\overline{\gamma} : \mathbb{R}^n \to \mathbb{R}$, given by

$$\overline{\gamma}(\theta) = E\left[c(\overline{P}_n, Q_n^-, Q_n)\right],$$

where the expectation is w.r.t. the stationary probability of the Markov chain $\{Q_n, \overline{P}_n\}$ of the state-action pairs.

The aim is to find a RSP that maximizes $\overline{\gamma}(\theta)$ over all θ. To this end we employ the actor-critic algorithm proposed in [8]. To deal with "state-space explosion" we employ an exponential approximation for the RSP μ_θ

$$\mu_\theta(x,\overline{p}) \quad \approx \quad \frac{e^{\sum_{j=1}^n \theta^i f^i(x,\overline{p})}}{\sum_{\overline{p}} e^{\sum_{j=1}^n \theta^i f^i(x,\overline{p})}}, \tag{1}$$

where $f^i(x,\overline{p})$ are the actor's features.

When the actor parameter vector is θ, the job of the critic is to compute an approximation to $\nabla\overline{\gamma}(\theta)$ which is then used by the actor to update its policy in an approximate gradient direction. As established in [17] this requires the computation by the critic of the projection of a certain quantity (denoted as q_θ in [8]) on the subspace spanned by the $\mathbb{R}^n$ valued function $\theta \to \Psi_\theta(x,\overline{p}) = \nabla \ln \mu_\theta(x,\overline{p})$ defined for each $(x,\overline{p}) \in \mathbb{X} \times \mathbb{P}$.

We again employ a linearly parameterized approximation architecture for this projection (calling it $Q^r_\theta(x,\overline{p})$)

$$Q^r_\theta(x,\overline{p}) = \sum_{j=1}^{m} r^j \phi^j_\theta(x,\overline{p}) \tag{2}$$

where $r = (r^1,\ldots,r^m) \in \mathbb{R}^m$ denotes the parameter vector of the critic and $\phi^j_\theta, j = 1,\ldots,m$ are the critic feature vectors.

Without going into further technicalities of the algorithm we next present the updation steps of actor and critic (which takes place in a single sample path simulation of the controlled Markov chain). Let $\hat{\gamma}$ be the scalar estimate of the average revenue and an m-vector $\hat{Z}$ be Sutton's eligibility trace [8]. At the kth iterate, let $r_k, \hat{Z}_k, \hat{\gamma}_k$ be the parameters of the critic, and let θ_k be the parameter vector of the actor. Let $\hat{X}_{k+1}$ be the new state, obtained after action $\hat{P}_k$ is applied. A new action $\hat{P}_{k+1}$ is generated according to the RSP corresponding to θ_k. Then:

- *Critic update:*

$$\begin{aligned}\hat{\gamma}_{k+1} &= \hat{\gamma}_k + b_k(c(\hat{P}_{k+1},\hat{X}^-_{k+1},\hat{X}_{k+1}) - \lambda_k),\\ r_{k+1} &= r_k + b_k d_k \hat{Z}_k,\end{aligned}$$

  where

$$\begin{aligned}d_k &= c(\hat{P}_k,\hat{X}^-_k,\hat{X}_k) - \hat{\gamma}_k + \\ &\quad +r_k{}'\phi_{\theta_k}(\hat{X}_{k+1},\hat{P}_{k+1}) - r_k{}'\phi_{\theta_k}(\hat{X}_k,\hat{P}_k)\end{aligned}$$

  and

$$\hat{Z}_{k+1} = \begin{cases}\hat{Z}_k + \phi_{\theta_k}(\hat{X}_{k+1},\hat{P}_{k+1}), & \text{if } \hat{X}_{k+1} \neq x^* \\ \phi_{\theta_k}(\hat{X}_{k+1},\hat{P}_{k+1}), & \text{otherwise}\end{cases}$$

  where x^* is any fixed state and a_k is a positive step-size parameter.

- *Actor update:*

$$\begin{aligned}\theta_{k+1} &= \theta_k - a_k\Gamma(r_k)r_k{}'\phi_{\theta_k}(\hat{X}_{k+1},\hat{P}_{k+1})\\ &\quad \Psi_{\theta_k}(\hat{X}_{k+1},\hat{P}_{k+1}),\end{aligned}$$

  where $\Gamma(.)$ is a scalar that controls the step-size a_k of the actor.

- The step sizes b_k and a_k are deterministic, non-increasing and satisfy

$$\sum_k b_k = \sum_k a_k = \infty,$$

$$\sum_k b_k^2 < \infty, \quad \sum_k a_k^2 < \infty \quad \text{and} \quad \sum_k \left(\frac{a_k}{b_k}\right)^d < \infty$$

  for some $d > 0$.

- The function $\Gamma(\cdot)$ is assumed to satisfy the following inequalities for some positive constants C_1 and C_2:

$$\frac{C_1}{1+\mid r \mid} \leq \Gamma(r) \leq \frac{C_2}{1+\mid r \mid}.$$

VI. Simulation Results

We show some preliminary simulation results for a simple two queue system with a single class of customers and linear disutility. More extensive simulations are being planned, including the multiclass, nonlinear disutility case.

The arrival process is assumed to be Poisson with rate $\lambda = 5$ and the two servers each have exponential service time with rate $\frac{\mu}{2}$ for $\mu = 4$. Let $\alpha_i = 8, B_i = 5$ and $\mathbb{P}^i = \{1,2,3,4,5\}$ for $i = 1,2$. (Thus the queues are symmetric.) Take $x^* = (1,1)$, initial queue lengths $Q(t_0) = (3,3)$ and initial price vector $p^0 = (2,4)$. The actor features are $f^i = x_i + p_i$ and $f^{i+2} = I(x_i + p_i < \alpha_i \ and \ x_i < B_i)$ for $i = 1,2$. The critic features are

$$\begin{aligned}\phi^j_\theta(x,\bar{p}) &= \frac{\partial ln\mu_\theta(x,\bar{p})}{\partial\theta^j}\\ &= f^j(x,\bar{p}) - \frac{\sum_{\bar{p}} f^j(x,\bar{p})e^{\sum_{i=1}^n \theta^i f^i(x,\bar{p})}}{\sum_{\bar{p}} e^{\sum_{i=1}^n f^i(x,\bar{p})\theta^i}},\end{aligned}$$

for $1 \leq j \leq 4$. We take

$$a_k = \frac{1}{k}, \quad b_k = \frac{1}{k^{0.85}}, \quad \Gamma(r_k) = \frac{100}{1+|r_k|}$$

The critic and actor weights r^i_k, θ^i_k respectively and the estimated average revenue γ_k is plotted in Figs. 1-3. In Fig. 4, the latter is plotted for four different values of balking thresholds, $(\alpha_1,\alpha_2) = (3,4),(5,5),(6,7),(8,8)$, respectively. It may be noted that the convergence is slow, which is typical of reinforcement learning algorithms, and is in tune with the well known dictum of adaptive control that there is a trade-off between learning speed and prior knowledge ('no free lunch').

VII. Conclusion

The round robin service of K parallel queues by a single server with differential join prices to each queue, that we call the Tirupati pricing scheme, is directly applicable to the provision of differential quality by an Internet service provider. We have shown that for a single class of customers with a linear disutility function this system is nearly socially optimal. It can be argued that the linear disutility is a reasonable assumption under a general model of congestion posting. If we consider either virtual circuit connections or long lived flows in the Internet the congestion information should be the number of active flows and in this case the bandwidth that will be seen by an arriving connection reduces in direct proportion to the number of active connections (or flows) in the system. Likewise, if the flows are short lived then the throughput performance, and

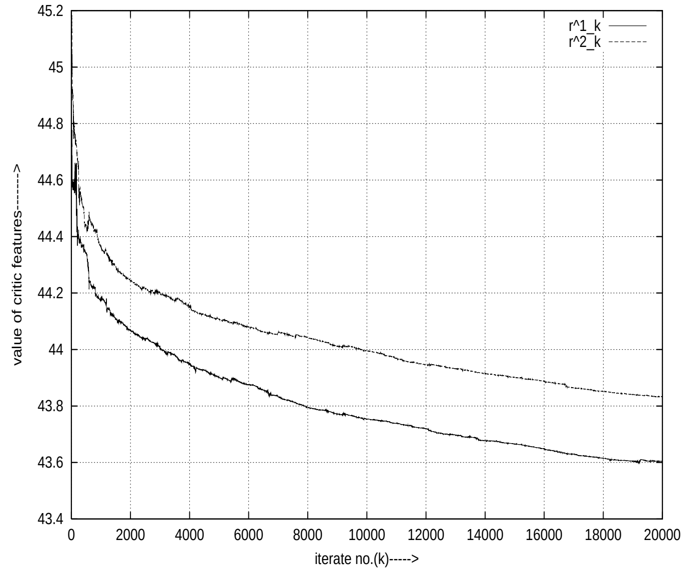

Fig. 1. Values of critic features vs. number of updates.

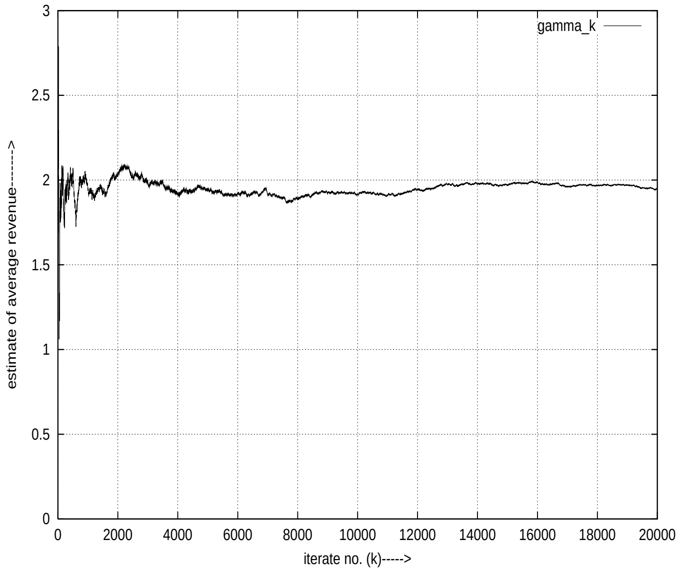

Fig. 3. Estimates of average revenue vs. number of iterates.

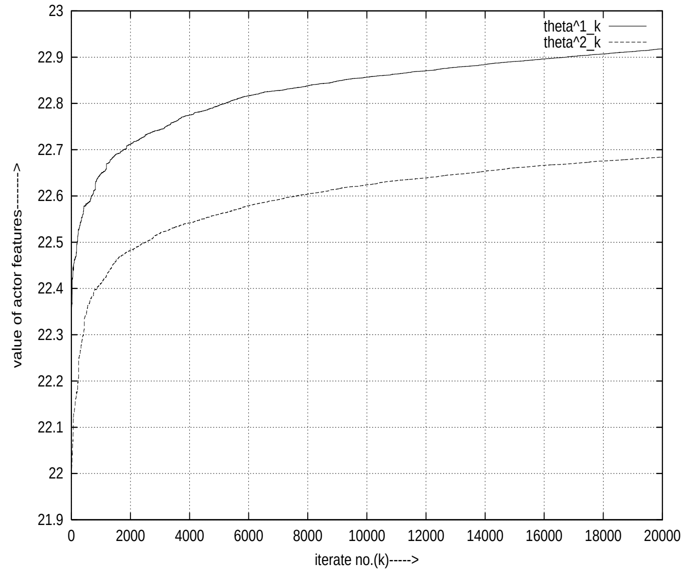

Fig. 2. Values of actor features vs. number of updates.

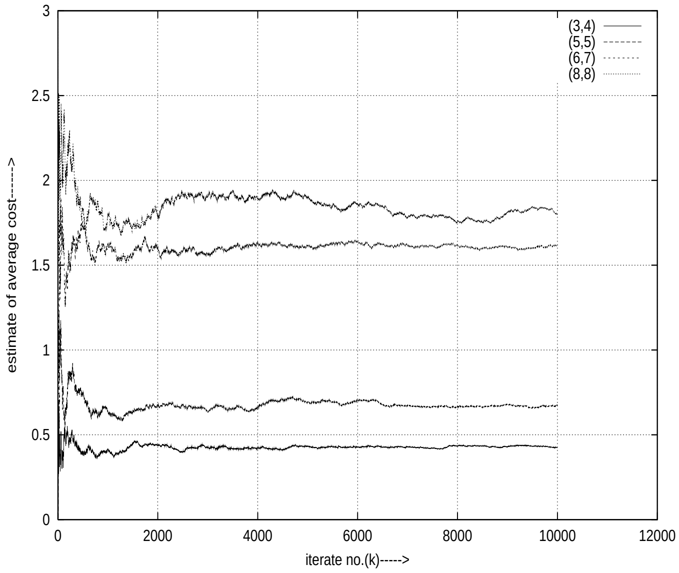

Fig. 4. Estimates of average revenue vs. number of iterates for different thresholds.

hence the response time, is directly affected by loss probability and, under a reasonable buffer management policy, this is determined by the number of packets in the queue. The congestion information for this case should be the number of packets in the queue.

The dynamic pricing algorithms that we have presented are fairly general and can consider multiclass customers and general disutility functions. An immediate future work involves a more detailed study of the performance of the algorithms in a multiclass customers environment with different disutility functions and different balking thresholds.

APPENDIX

We prove here Lemma 1 2. Thus a customer arriving at time t finds the i-th queue blocked if $Q_i(t^-) > \beta_i = \min(B_i+p_i, \alpha_i)$ and joins the minimum length unblocked queue if any, balks otherwise. We shall show that in this case, JSQ minimizes the average total queue length

$$\limsup_{t\to\infty} \frac{1}{t}\int_0^t E[\sum_{i=1}^K Q_i(s)]ds.$$

The proof will closely mimic the 'forward induction' argument of Section 8.3, [15], with two important differences. The first is that we allow for balking. Secondly, we fill in some subtle details that seem to have been glossed over in [15]. The analysis uses the following partial ordering of $K-$dimensional nonnegative random variables : Given $\mathbf{x} = [x_1, \cdots, x_K],\ x_i \geq 0$, define $H_i(\mathbf{x}) = \sum_{j=1}^i x_{k_j}$ for $1 \leq i \leq K$, where $[x_{k_1}, \cdots, x_{k_K}]$

is a permutation of $[x_1, \cdots, x_K]$ arranged in a nonincreasing order. Given $K-$dimensional nonnegative random variables $\mathbf{X}$, $\mathbf{Y}$, we say that $\mathbf{X} b \mathbf{Y}$ if there exist random variables $\bar{\mathbf{X}}, \bar{\mathbf{Y}}$ on a common probability space such that $\mathbf{X}$, $\bar{\mathbf{X}}$ (respectively, $\mathbf{Y}$, $\bar{\mathbf{Y}}$) agree in law and

$$P(H_i(\bar{\mathbf{Y}}) \geq H_i(\bar{\mathbf{X}}),\ 1 \leq i \leq K) = 1.$$

The following is then proved as in [15], pp. 262.

Lemma 3: $\mathbf{X} b \mathbf{Y}$ implies $E[\sum_{j=1}^{i} Y_i] \geq E[\sum_{j=1}^{i} X_i]$, $1 \leq i \leq K$.

Consider now two realizations of the $\mathbf{Q}(\cdot)$ process, denoted by $\mathbf{X}(\cdot)$, $\mathbf{Y}(\cdot)$ respectively, on a common probability space such that the arrival and departure instants as well as the initial condition are identical for both, $\mathbf{X}(\cdot)$ obeys JSQ, and $\mathbf{Y}(\cdot)$ is governed by some other policy. In particular, $\mathbf{X}(0) b \mathbf{Y}(0)$. Furthermore, it is assumed that when there is a potential departure from the i-th largest queue in the former, it is also so in the latter. As pointed out in [15], this affects the joint distribution of the two processes, but not their individual law. We shall show that $\mathbf{X}(t) b \mathbf{Y}(t)$ for all t, which in turn implies the claim in view of Lemma 3. To prove this, we assume that $\mathbf{X}(t) b \mathbf{Y}(t)$ for some $t_{n-1} \leq t < t_n$, $n \geq 1$, and prove that $\mathbf{X}(t_n) b \mathbf{Y}(t_n)$. We need to consider several different cases.

Case 1: Departures from both systems

Suppose there is a departure from the k-th ranked queue in both $\mathbf{X}(\cdot)$ and $\mathbf{Y}(\cdot)$ and that it makes the queue move down in rank by $m_1 \geq 0$ places in the former and $m_2 \geq 0$ places in the latter. This implies that the queues ranked $k, k+1, \cdots, k+m_1$ have equal lengths in $\mathbf{X}(\cdot)$ and likewise those ranked $k, k+1, \cdots, k+m_2$ have equal lengths in $\mathbf{Y}(\cdot)$. It suffices to consider $m_1 \neq m_2$ as clearly $\mathbf{X}(t_n) b \mathbf{Y}(t_n)$ when the two are equal.

- $m_1 < m_2$: For $i < k + m_1$,

  $$H_i(\mathbf{Y}(t_n)) = H_i(\mathbf{Y}(t)) \geq H_i(\mathbf{X}(t)) = H_i(\mathbf{X}(t_n)).$$

  For $i \geq k + m_2$,

  $$H_i(\mathbf{Y}(t_n)) = H_i(\mathbf{Y}(t)) - 1 \geq H_i(\mathbf{X}(t)) - 1 = H_i(\mathbf{X}(t_n)).$$

  For $k + m_1 \leq i < k + m_2$,

  $$H_i(\mathbf{Y}(t_n)) = H_i(\mathbf{Y}(t)) > H_i(\mathbf{X}(t)) - 1 = H_i(\mathbf{X}(t_n)).$$

  Thus $\mathbf{X}(t_n) b \mathbf{Y}(t_n)$.
- $m_1 > m_2$: As above, $H_i(\mathbf{Y}(t_n)) \geq H_i(\mathbf{X}(t_n))$ for $i < k + m_2$ and $i \geq k + m_1$. Suppose for some $k + m_2 \leq i < k + m_1$, $H_i(\mathbf{Y}(t_n)) = H_i(\mathbf{Y}(t)) - 1 < H_i(\mathbf{X}(t_n)) = H_i(\mathbf{X}(t))$. Since $H_i(\mathbf{Y}(t)) \geq H_i(\mathbf{X}(t))$, we must have $H_i(\mathbf{Y}(t)) = H_i(\mathbf{X}(t))$. Also, since $H_{i+1}(\mathbf{Y}(t)) \geq H_{i+1}(\mathbf{X}(t))$, we have $\tilde{Y}_{i+1}(t) \geq \tilde{X}_{i+1}(t)$ where we use $\tilde{Y}_j(s)$ (resp., $\tilde{X}_j(s)$) to denote the $j-$th ranked queue in $\mathbf{Y}(s)$ (resp., $\mathbf{X}(s)$) for $s \geq 0$. Thus

  $$\tilde{Y}_i(t) \geq \tilde{Y}_{i+1}(t) \geq \tilde{X}_{i+1}(t) = \tilde{X}_i(t).$$

  If $\tilde{Y}_i(t) > \tilde{X}_i(t)$, then for $H_i(\mathbf{Y}(t)) = H_i(\mathbf{X}(t))$ to hold, we must have $H_{i-1}(\mathbf{Y}(t)) < H_i(\mathbf{X}(t))$, a contradiction. Thus $\tilde{Y}_i(t) = \tilde{X}_i(t)$ and $H_{i-1}(\mathbf{Y}(t)) = H_{i-1}(\mathbf{X}(t))$. Repeating this argument, one can show that $\tilde{Y}_j(t) = \tilde{X}_j(t)$, $H_j(\mathbf{Y}(t)) = H_j(\mathbf{X}(t))$ for $k + m_2 \leq j \leq i$. But

  $$\begin{aligned} H_{k+m_2+1}(\mathbf{Y}(t)) &< H_{k+m_2}(\mathbf{Y}(t)) \\ &= H_{k+m_2}(\mathbf{X}(t)) \\ &= H_{k+m_2+1}(\mathbf{X}(t)). \end{aligned}$$

  The inequality above follows because the $k-$th ranked queue in $\mathbf{Y}(t)$ moved to $(k+m_2)-$th place at t_n, the first equality is proved above and the second equality follows from the fact that $m_1 > m_2$. Thus $H_{k+m_2+1}(\mathbf{Y}(t)) < H_{k+m_2+1}(\mathbf{X}(t))$, a contradiction. Thus $H_i(\mathbf{Y}(t_n)) \geq H_i(\mathbf{X}(t_n))$ for all i.

Case 2: Departure from only one system

In the case where at time t_n there is a departure from the k-th ranked queue of $\mathbf{X}(\cdot)$ but none from the $k-$th ranked queue of $\mathbf{Y}(\cdot)$ (because $\tilde{Y}_k(t) = 0$), it is easy to see that $\mathbf{X}(t_n)) b \mathbf{Y}(t_n))$. Thus consider the case when $\tilde{X}_k(t) = 0$ and there is a departure from the $k-$th ranked queue in $\mathbf{Y}(t)$ which moves down to rank $k + m$, $m \geq 0$. This will be the case when $\tilde{Y}_i(t) = \tilde{Y}_k(t)$ for $k \leq i \leq k + m$. Thus $H_i(\mathbf{Y}(t_n)) \geq H_i(\mathbf{X}(t_n))$ for $i < k + m$. Suppose for some $i \geq k + m$, $H_i(\mathbf{Y}(t_n)) = H_i(\mathbf{Y}(t)) - 1 < H_i(\mathbf{X}(t)) = H_i(\mathbf{X}(t_n))$. Since $H_i(\mathbf{Y}(t)) \geq H_i(\mathbf{X}(t))$, we must have $H_i(\mathbf{Y}(t)) = H_i(\mathbf{X}(t))$. But $H_{i-1}(\mathbf{Y}(t)) \geq H_{i-1}(\mathbf{X}(t))$, thus $\tilde{Y}_i(t) \leq \tilde{X}_i(t) = 0$, leading to $\tilde{Y}(t_n)) = -1$, a contradiction. Thus $\mathbf{X}(t_n) b \mathbf{Y}(t_n)$ must hold.

Case 3: Arrival without blocking

If no queue is blocked for an arrival at t_n in either $\mathbf{X}(\cdot)$ or $\mathbf{Y}(\cdot)$, it will join the $K-$th ranked queue in the former and $k-$th ranked queue in the latter for some $k \leq K$. It is then easy to see as in [15], p. 263, that $\mathbf{X}(t_n) b \mathbf{Y}(t_n)$.

Case 4: Arrival with blocking

If $\mathbf{X}(t) b \mathbf{Y}(t)$ and an arrival at t_n balks from $\mathbf{X}(\cdot)$, it is easy to see that $\mathbf{X}(t_n) b \mathbf{Y}(t_n)$ no matter what. Thus we only need consider the case when some but not all queues in $\mathbf{X}(\cdot)$ are blocked for the arriving customer. Also, suppose that in either of the processes the arriving customer joins $j-$th ranked queue, which then moves up to $k-$th rank for some $k \leq j$. Then it must be that $k-$th, $\cdots$, $j-$th queues had equal lengths and $(k-1)-$st queue had a strictly higher length. Thus without loss of generality, we may assume that when the arrival joins one of a set of queues with identical lengths, it is always the one that has been ranked the highest and therefore the ranking does not change.

Now suppose the arriving customer joins $j-$th queue in $\mathbf{X}(\cdot)$ and $k-$th queue in $\mathbf{Y}(\cdot)$ for some $k \leq j$. Then $H_i(\mathbf{X}(t_n)) = H_i(\mathbf{X}(t))$ for $i < j$ and $= H_i(\mathbf{X}(t)) + 1$ for $i \geq j$, similarly for $H_i(\mathbf{Y}(t_n))$ with k replacing j. Since $k \leq j$, it follows that $H_i(\mathbf{X}(t_n)) b H_i(\mathbf{Y}(t_n))$.

Next suppose that $k > j$. Suppose that for some $i, j \leq i < k$, we have

$$H_i(\mathbf{X}(t_n)) = H_i(\mathbf{X}(t)) + 1 > H_i(\mathbf{Y}(t_n)) = H_i(\mathbf{Y}(t)).$$

Since $H_i(\mathbf{X}(t)) \leq H_i(\mathbf{Y}(t))$, we must have $H_i(\mathbf{X}(t)) = H_i(\mathbf{Y}(t))$. Now there are two possibilities: Either (i) $\tilde{X}_{i+1}(t) = \tilde{X}_j(t)$ or (ii) $\tilde{X}_{i+1}(t) < \tilde{X}_j(t)$. Consider the former case. Argue as in *Case 1* above with $m_1 > m_2$ to conclude that $\tilde{X}_\ell(t) = \tilde{Y}_\ell(t)$, $H_\ell(\mathbf{X}(t)) = H_\ell(\mathbf{Y}(t))$ for all $j \leq \ell \leq i$.

Since $H_{i+1}(\mathbf{X}(t)) \leq H_{i+1}(\mathbf{Y}(t))$, we must have $\tilde{X}_{i+1}(t) \leq \tilde{Y}_{i+1}(t) \leq \tilde{Y}_i(t) = \tilde{X}_i(t)$. Thus if $\tilde{X}_i(t) = \tilde{X}_{i+1}(t) = \tilde{X}_j(t)$, then $\tilde{Y}_i(t) = \tilde{Y}_{i+1}(t) = \tilde{X}_i(t)$ and $H_{i+1}(\mathbf{X}(t_n)) > H_{i+1}(\mathbf{Y}(t_n))$. Repeating this, if $\tilde{X}_r(t) = \tilde{X}_j(t)$ for $i \leq r \leq \ell$, $\tilde{X}_r(t) = \tilde{Y}_r(t)$ and therefore $H_r(\mathbf{X}(t)) = H_r(\mathbf{Y}(t))$, $H_r(\mathbf{X}(t_n)) > H_r(\mathbf{Y}(t_n))$ for $i \leq r \leq \ell$. Hence we may replace i by ℓ if necessary and suppose that $\tilde{X}_{i+1}(t) < \tilde{X}_i(t)$, which is the case (ii) above. In this case it must be that $\tilde{X}_r(t) = \beta_{j_r}$ for $i < r \leq K$, where j_r is such that $\tilde{X}_r(t) = X_{j_r}(t)$. Then we must have $\tilde{Y}_s(t) \geq \beta_{j_s}$ for $s \geq r > i$. Then queues ranked $(i+1)$ onwards for $\mathbf{X}(t)$ would also have ranks $(i+1)$ on for $\mathbf{Y}(t)$, in the same order as for $\mathbf{X}(t)$, i.e., in the order of decreasing β_{j_r}'s. It then follows that these will be blocked for $\mathbf{Y}(t)$ as well. This contradicts the fact that $k > j$. Thus $\mathbf{X}(t_n) b \mathbf{Y}(t_n)$ must hold.

The case when the customer joins the $j-$th queue in $\mathbf{X}(\cdot)$ and balks in $\mathbf{Y}(\cdot)$ is easy. Since the customer balks in $\mathbf{Y}(\cdot)$, $\tilde{Y}_i(t) = Y_{k_i}(t) = \beta_{k_i}$, where $\{k_1, \cdots, k_K\}$ is a permutation of $\{1, \cdots, K\}$ such that $\beta_{k_1} \geq \beta_{k_2} \geq \cdots \geq \beta_{k_K}$. Suppose for some $i \geq j$, $H_i(\mathbf{X}(t_n)) > H_i(\mathbf{Y}(t_n))$. Then as above, $H_i(\mathbf{X}(t)) = H_i(\mathbf{Y}(t)) = \sum_{s=1}^{i} \beta_{k_s}$, which is possible only if $\tilde{X}_s(t) = \beta_{k_s}$ for $1 \leq s \leq i$. Thus all queues up to $i-$th, in particular the $j-$th queue, are blocked for $\mathbf{X}(\cdot)$ at time t_n^-, a contradiction.

References

[1] J. Abounady, D. Bertsekas and V. S. Borkar, "Learning Algorithms for Markov Decision Processes with Average Cost," *SIAM J. Control and Optim.*, to appear.

[2] F. Baumgartner, T. Braun and F. Habegger, "Differentiated Services: A New Approach for Quality of Service in the Internet", in H. van As (ed.): High Performance Networking, Kluwer, 1998,ISBN: 0-412-84660-8, pp. 255-274.

[3] S. Blake, *et al*, "An Architecture for Differential Services," *IETF RFC 2475,* Dec. 1998.

[4] D. P. Bertsekas, "Dynamic Programming and Optimal Control", vol. 2, *Athena Scientific 1995*

[5] I. C. Paschalidis, J. N. Tsitsiklis, "Congestion-Dependent Pricing of Network Services", *IEEE/ACM Trans. Networking*, vol. 8, no. 2, pp. 171-184, Apr. 2000.

[6] M. Currence, A. Kurzon, D. Smud and L. Trias, "A Causal Analysis of Usage-Based Billing on IP Networks", URL: *citeseer.nj.nec.com/currence00causal.html*

[7] A. Ephremides, J. Walrand and P. Varaiya, "On the Optimality of Join Shortest Queue Policy," *IEEE Transactions on Communications,* 1987.

[8] V. R. Konda and J. N. Tsitsiklis, "Actor-Critic Algorithms", submitted to *SIAM Journal on Control and Optimization*, Feb 2001, available at *http://www.mit.edu/people/jnt/publ.html.*

[9] M. Falkner, M. Devetsikiotis, I. Lambadaris, "An Overview of Pricing Concepts for Broadband IP Networks, *IEEE Communications Surveys*, Sept. 2000, pp. 2-13.

[10] F. Kelly, "Charging and Rate Control for Elastic Traffic," *European Transactions on Telecommunication,* vol 8, 1997, pp 33-37.

[11] V. R. Konda, V. S. Borkar, "Actor-Critic-Type Learning Algorithms for Markov Decision Processes," *SIAM J. Control and Optim.* 38(1), 1999, pp. 94-123.

[12] R. D. Foley and D. R. McDonald, "Join Shortest Queue: Stability and Exact Asymptotics", submitted to *Annals of Applied Probability*, available at *http://www.isye.gatech.edu/ rfoley/pub.html.*

[13] A. Odlyzko, "Paris Metro Pricing for the Internet", in the *Proc. of ACM Conf. on Electronic Commerce*, 1999, pp. 140-147.

[14] M. I. Puterman, *Markov Decision Processes*, John Wiley, New York, 1994.

[15] J. Walrand, *Introduction to Queueing Networks*, Prentice Hall, Englewood Cliffs, NJ, 1988.

[16] J. Walrand and P. Varaiya, *High Performance Communication Networks,* 2 Ed, Morgan Kaufman, 1999.

[17] P. Marbach and J. N. Tsitsiklis, "Simulation-Based Optimization of Markov Reward Processes", *IEEE Trans. Automatic Control* 46(2), Feb., 2001, pp. 191-209.