DiffServ Node with Join Minimum Cost Queue Policy: Analysis with Multiclass Traffic

D. Manjunath
Dept. of Electrical Engg
Indian Institute of Technology, Bombay Powai Mumbai 400076 INDIA
dmanju@ashgo@ee,nh@iitb.ac.in

Abstract—DiffServ is an attractive candidate for providing relative QoS in the Internet. This is also easily amenable to simple and effective pricing mechanisms. By pricing access to a relative QoS, we can model a DiffServ node as a “Join Minimum Cost Queue” in which an arriving customer (packet or connection) determines the relative cost as a function of the congestion in the different queues and their access prices and decides to take service from that queue for which the cost is minimum. The Paris Metro pricing system and its work conserving variant called the Tirupati pricing are analyzed in this paper in the presence of multiclass traffic and for static pricing. Two of the more interesting observations are that the delay and revenue rate are not monotonic or convex functions of price and the revenue rate is very sensitive to the behavior of the delay sensitive class.

I. INTRODUCTION

Many pricing models to save the Internet from suffering the “tragedy of commons” have been proposed. Considering the volume (in bytes, packets or connections) of traffic handled by the Internet, it is now clear that these schemes should have simple implementations and be robust. [1] identifies four categories of Internet charges -- access, usage, congestion and quality. It is also generally believed that in this classification, it is computationally expensive to implement pricing structures for all but the access charges. [2] is an good overview and bibliography on Internet pricing issues.

Absolute end-to-end quality assurances is very difficult and probably infeasible in the Internet. Relative guarantees on a hop-by-hop basis are more feasible. The DiffServ [3] model for QoS in the Internet is based on this realisation. In DiffServ the available bandwidth is divided among multiple classes statically or dynamically. Network nodes maintain separate logical queues for each class for each outgoing link and service them according to the bandwidth sharing policy. Priority queues are the simplest way to provide multiclass service but it could lead to starvation of lower priority queues for extended periods of time. A second option would be to have multiple queues and force a certain arrival profile to each queue to provide different QoS guarantees to arrivals to that queue. Pricing seems to be a good vehicle to provide this type of control. Paris Metro Pricing system (PMP) is one such pricing system for multiclass queues [4]. In the PMP, the network is logically partitioned in a static manner, each partition is allocated a fixed share of the network resources and the access to each partition is differentially priced. In [5], [6] the behavior of PMP under equilibrium conditions is considered and compared with a uniclass pricing system.

[7] proposes a work conserving version of PMP called the Tirupati system. This takes inspiration from the queue management system at Tirupati, a major pilgrimage center in southern India where it has been operating with remarkable efficiency for quite some time now. In [7], the social optimality of Tirupati pricing is analyzed and it was shown that the difference between the social cost of the optimally priced system and that of the Tirupati system is $K^*$ for constants $K$ and $c$. A more practical contribution of [7] was dynamic pricing using a dynamic programming equation and a reinforcement learning based online pricing algorithm.

In this paper we consider pricing quality through an access charge and analyze pricing in a multiclass service system like DiffServ where the access charge is a function of the service class. We abstract an Internet node as a queuing system and analyze the Tirupati and PMP systems with multiclass traffic and static pricing. Our results are from classical Markovian queueing analysis similar to the analysis of joint shortest queue systems from where we borrow some techniques. The paper is organized as follows. In the next section we describe our models and assumptions. Section III describes the analysis methods to obtain revenue rate bounds and disutility rate (to be defined later) approximations for an infinite buffer system. Section IV presents numerical results from our analysis.

II. JOIN MINIMUM COST QUEUE WITH MULTICLASS CUSTOMERS

The “Join Minimum Cost Queue (JMCoQ)” works as follows. A single exponential server of capacity $\mu$ serves $K > 1$ queues. The $K$ queues have different join prices $p_1 \leq p_2 \leq \cdots \leq p_K$. We assume Poisson arrivals at rate $\lambda$. Arrivals belong to $M$, $M \geq 1$ classes, and an arrival chooses class $m$ with $r_m$ independent of previous arrivals. Let $N_i(t)$ be the number of customers in queue $i$, $1 \leq i \leq K$ and let $N(t) = [N_1(t), N_2(t), \ldots, N_K(t)]^T$ be the queue length vector at time $t$. A class $m$ arrival at time $t$ is informed of $N(t^-)$, the queue vector 'just before $t$'. It calculates its disutility for queue $i$ according to a function $\psi(m, N_i(t), p_i)$ and joins...
but do not pursue them in this paper.

Thus the disutility function that involve a model of customer behavior based upon the disutility rate for class m customers, \( \psi_m \). We assume that the customers obtain instantaneous congestion information on arrival and that there is no delay as might be expected in the context of an Internet. An important contribution of this paper is that we explicitly involve a model of customer behavior based upon a "disutility" calculated from the posted prices and congestion levels. We can also consider queuing systems and finite buffer queues but do not pursue them in this paper.

A convenient combination of the queue length (the congestion indicator) and price as a disutility function is simple and effective in capturing price and delay sensitivities for different types of traffic. Thus the disutility function that we consider is \( \psi(m, N_j(t), p_m) = \alpha_m N_j(t) + (1 - \alpha_m)p_m \), with \( \alpha_m \in (0, 1) \). The \( \alpha_m \) will be called the delay sensitivity of class m. We further assume that the service rates for all classes are identical and that prices are not class dependent, a reasonable assumption in the context of the Internet. Therefore, \( \alpha_m = \alpha_m \) for \( m = 1, \ldots, M \) and \( p_m = p \) for \( i = 1, \ldots, K \) and \( \psi() \) reduces to \( \psi(m, N_i(t), p) = \alpha_m N_i(t) + (1 - \alpha_m)p \).

In the sequel we consider a balanced JMCQ with two customer classes and two queues, \( M = 2 \) and \( K = 2 \). We will also assume that \( \mu_1 = \mu_2 = \mu/2 \). Without loss of generality, we assume \( \alpha_1 > \alpha_2 \), i.e., class 1 traffic is delay sensitive and class 2 traffic is price sensitive. In the next section we analyze the system with infinite buffers and no balking.

### III. Analysis of Infinite Buffer JMCQ

The JMCQ policy is similar to the "Join Shortest Queue" (JSQ) policy that has been extensively studied. The exact performance analysis of the JSQ system has been notoriously difficult and only approximations and asymptotic results are available for the general problem. See [8] for a survey of the analytical models for JSQ. The JMCQ is further complicated by the existence of a cost associated with the queues and also the presence of multiple classes of customers and we expect that exact closed form expressions for performance analysis will be difficult. Therefore, we consider an approximate analysis. We will concentrate on two kinds of results - the revenue rate for the server and the disutility rate for the customers.

The evolution of \( N(t) \) is a two dimensional birth death process. Given \( N(t) \), the departure process is governed by the service policy - non work conserving PMP or work conserving Tirupati. The queue in which an arrival state \( N(t) \) will join is determined by its class and its delay sensitivity. Let \( \delta_{ij}^m \) be the queue that an arriving class m customer to state \( [i, j] \) will join. For example, \( \delta_{ij}^1 = 1 \) if \( \psi(1, i, p_1) \leq \psi(1, j, p_2) \) and \( \delta_{ij}^2 = 2 \) otherwise. The transition rate matrix \( Q = \{ q_{ij}; \} \), \( i, j, k = 0, 1, \ldots \) is easily determined.

We assume that the Markov chain described by \( Q \) above is ergodic and a steady state solution \( \pi = \{ \pi_{ij} \} \) exists. When the system is in state \( [i, j] \), a revenue \( \pi_i \) is earned if an arrival joins queue i. Therefore, from the Law of Large Numbers, the steady state rate of revenue generation, \( R \), is given by [9]

\[
R = \sum_{i,j} \pi_{ij} \lambda [r_1 p_1 I(\delta_{ij}^1 = 1) + r_2 p_2 I(\delta_{ij}^1 = 2) +
+r_2 p_1 I(\delta_{ij}^2 = 1) + r_1 p_2 I(\delta_{ij}^2 = 2)]
\]

(1)

where \( I(\cdot) \) is the indicator function with the usual meaning of taking on a value 1 if the condition is satisfied. Similarly, the disutility rate for class m customers, \( D_m \), is given by

\[
D_m = \sum_{i,j} \pi_{ij} \lambda r_m [I(\delta_{ij}^m = 1)(a_m i + (1 - a_m)p_1) +
+I(\delta_{ij}^m = 2)(a_m j + (1 - a_m)p_2)]
\]

(2)

We now describe the method to obtain bounds and approximate results. First consider the revenue rate, \( R \). We borrow the techniques of [10] to obtain bounds on \( R \) by considering a truncated state space. To motivate the truncation, first consider the situation when only one class of customers is present. Without loss of generality, let \( p_1 < p_2 \). On the \( N_1(t) = N_2(t) \) plane, an attractor line can be defined such that an arrival to the system when the state is on the left of this line will join queue-1 and an arrival to a state on the right of this line will join queue-2. This means that arrivals tend to move the system towards the attractor. The attractor line is defined by the equation \( N_2 = N_1 - (\frac{1 - a_m}{a_m})(p_2 - p_1) \), i.e., arrivals will prefer the lower priced queue unless \( N_1 - N_2 > (\frac{1 - a_m}{a_m})(p_2 - p_1) \).

With multiclass traffic, there will be separate attractors for each class. Therefore, for states \( (i, j) \) away from the attractor we argue that \( \pi_{ij} \) becomes very small and we can consider a state space truncated at some distance from the attractors and suitably modified transition rates. Specifically,
we will consider the state space of the JMCQ truncated to contain only those states for which 0 ≤ |N₁ - N₂| < T, T ≥ (1 - α₂ + 1 - α₁)(p₂ - p₁). T will be called the truncation threshold. Let S denote the state space of the original JMCQ and S', the state space obtained after truncation.

We define systems JMCQ₁, JMCQ₂ over S', like in [10]. These give us upper and lower bounds on the mean delay and the mean number in the system for the JMCQ. First consider system JMCQ₁ and let Q₁ = (q₁(i,j;kl)) be its transition rate matrix which is obtained from Q as follows. The transition rates from the states that are between the two truncation threshold lines will be the same as in the JMCQ system. In the states on the truncation threshold line, an arrival will move the state space towards its attractor and will not cause the system to go out of S'. In a state on the threshold line, a departure is disallowed. This means,

\[ q_{ij;kl}^{(u)} = 0 \quad \forall i_j \in S' \]

Now consider system JMCQ₂ and its transition rate matrix Q₂ = (q₂(i,j;kl)) which is obtained from Q as follows. As with JMCQ₁, the transition rates from the states between the threshold lines are the same as that in the JMCQ system. Once again, as with the JMCQ₁, we need to consider only the transition rates corresponding to departures from states on the truncation threshold line. These are modified such that a departure will take away a customer from the other queue. This means,

\[ q_{ij;kl}^{(l)} = \begin{cases} 0 & \text{if } i_j \text{ is on left threshold; } k = i - 1, l = j - 1 \\ q_{ij;kl} & \text{otherwise} \end{cases} \]

Let \( D_{ml} \) and \( \overline{N}_u \) be the corresponding quantities for the JMCQ系统的第l个样本。

**Proposition 1:**
1. \( D_{ml} \leq D_m \leq D_{mu} \)
2. \( \overline{N}_l \leq \overline{N}_i \leq \overline{N}_u \)

The proof is exactly similar to that in [10].

Let \( S_T \) be the states on the left threshold line and \( S_r \) the states on the right threshold line. Let \( \pi_{ij}^l \) be the stationary distribution of JMCQ₁ and define \( R_u \) as

\[ R_u = \sum_{i_j \in S'} \{ \pi_{i_j}^l [r_1 p_1 I(\delta_{ij} = 1) + r_1 p_2 I(\delta_{ij} = 2)] + \sum_{i_j \in (\delta_{ij} < 0)} \pi_{i_j}^l \mu p_1 + \sum_{i_j \in (\delta_{ij} > 0)} \pi_{i_j}^l \mu p_2(3) \} \]

Along similar lines we can define \( R_r \).

**Proposition 2:** \( R_u \geq R \geq R_r \)

**Proof:** We sketch the proof of the first half of the above inequality. The JMCQ₁ and JMCQ₂ behave identically to the JMCQ for states between the two threshold lines. Further, when the state is on a threshold line, the JMCQ₁ and JMCQ₂ behave identically to JMCQ for arrivals and we need to consider only departures corresponding to transitions \( d_1 \) and \( d_2 \) of Figure 1.

- Disallowing transition \( d_1 \) increases the disutility of the lower priced queue leading subsequent arrivals to join the higher priced queue with a higher probability, hence increasing the revenue on an average. While the system is on the left threshold line, revenue \( p_1 \) is accumulated into \( R_u \) at rate \( \frac{\mu}{2} \) in Eqn. 3.
- Similarly, \( d_2 \) is disallowed in JMCQ₂ and from Eqn. 3, \( R_u \) accumulates revenue \( p_2 \) at rate \( \frac{\mu}{2} \) while the system is on the right threshold line. This more than compensates for the loss in revenue due to increase in disutility of queue 2.

The JMCQ₁ and JMCQ₂ systems can be solved as follows. Rename the states in \( S' \) as follows: \((N₁, N₂)\) is renamed as \((N₁ - N₂, N₂)\) for \( N₁ = N₂ = \ldots = N₂ + T \). Since the system is a two-dimensional birth death process, the non zero transition rates are only between "adjacent" states and the transition rate matrix for the JMCQ₁ and JMCQ₂ systems will have the block-tridiagonal structure of a quasi birth death process and the matrix geometric techniques [11] are directly applicable. For more details of this mapping see [10].

**IV. NUMERICAL RESULTS**

We present some numerical results of our analysis of the PMP and Tirupati variations of the JMCQ and analyze the dependence of revenue and disutilities on the delay sensitivities of the customer types and the price vector. We will use \( p_1 = 0 \) and \( p_2 = p \). Thus price variation will essentially mean a variation of \( p \). Also, unless explicitly mentioned, \( \lambda_1 = \lambda_2 = 0.5 \delta \), the arrivals are equally likely to belong to either traffic class.
Further, the results are primarily for the Tirupati case unless explicitly mentioned.

First, consider the tightness of the bounds on the revenue rate. Figure 2 shows the upper and lower bounds on the revenue rate. The bounds are very tight, especially for higher values of $p$. Observe that although the revenue rate generally increases with increasing $p$, it does not have a unique maximum. This means that $R$ does not have a unique maximum.

Note that we have not established that the disutility rates obtained from either the JMCQ$_0$ or the JMCQ$_1$ systems are bounds on the actual disutility. We will use the JMCQ$_0$ system to obtain the disutility rates and treat the results as an approximation of the actual value rather than as an upper bound.

Next we compare the revenue rate in the Tirupati and PMP pricing models. In PMP, if $\lambda > \mu_1$, queue 1 will never empty and the system will always operate near the equilibrium lines. Increasing $p$ will only move the equilibrium lines and $R$ will be an increasing function of $p$. This is shown in Figure 3 for $\lambda = 0.7$. However, when $\lambda \leq \mu_1$, i.e., $\lambda = 0.4$, the revenue rate function of PMP looks similar to that of the Tirupati.

We now consider the effect of price sensitivity on $R$ and $D_m$ in the Tirupati queue. Figures 4 and 5 show the revenue rate as a function of the delay sensitivities of the two classes of traffic. Although $R$ increases with increasing $a_m$, the following observations are interesting.

1. The change in the revenue rate due to change in $a_1$ is much more than that caused by a change $a_2$. For example, even doubling of $a_2$ from 0.3 to 0.6 does not change the revenue rate appreciably.
2. The price at which the maximum revenue rate is achieved is very nearly the same for the different values $a_2$ whereas it increases for increasing values of $a_1$.

This is because a large fraction of the revenue is derived from the delay sensitive class and any changes in its behavior have a more noticeable effect than those in the price sensitive class. We also observed that the revenue is more sensitive to the delay sensitive class even when $(a_1 - a_2)$ is small.

Figure 6 shows the disutility rate as a function of $p$ for different $a_1$ and $a_2$ for Tirupati and PMP systems. Increasing $a_1$ decreases the disutility rate while increasing $a_2$ increases it. Further, it is an increasing function of $p$. However, as the price increases, the disutility rate does not increase beyond a certain value. This is because all arrivals join queue 1 and therefore the disutility is less and less a function of $p$. For PMP, the disutility increases much more rapidly and linearly.

To study revenue rate as a function of $\lambda$, we consider two cases - (i) $\lambda$ is varied with $\lambda_1 = \lambda_2 = 0.5 \lambda$, (ii) $\lambda$ is constant and $\lambda_1$ and $\lambda_2$ are varied. Figure 7 shows that the revenue rate behaves much like the mean delay function as a function of $\lambda$. This is expected because at high arrival rates, queue lengths can be significant and the delay sensitive traffic will choose...
V. DISCUSSION

We have analysed a DiffServ node with multiclass traffic operating as a join minimum cost queue. Specifically, we have analysed the tradeoffs between a service provider's gain (revenue) and the customers' cost (disutility function of price and delay). Although our disutility function is simple, the qualitative conclusions are fairly general and not expected to vary significantly for more complex disutility functions. Specifically, our observation of the significant sensitivity of the revenue to the behavior of the delay sensitive traffic.

We are currently investigating the system performance in the presence of balking traffic and finite resources at the queue. We are also considering more than two traffic classes and more sophisticated disutility functions.

ACKNOWLEDGMENTS

We thank Rahul Tandra for discussion on the proofs and for help in computation of the bounds.

REFERENCES


