Analysis of Shared Memory Priority Queues with two levels of traffic

Shlomi Bergida and Yuval Shavitt.

***Abstract*— Two rate SLAs become incrisingly popular in todays Internet, allowing a customer to save money by paying one price for commited traffic and a much lower price for additional traffic which is not guaranteed. These type of SLAs are suggested for all types of traffic from best effort to QoS constraint applications. Dimensioning and management of queues for multiple priorities each with two levels of guarentees becomes an interesting challange.**

We present a simple analysis of a multipriority multi discard level system controlled by a buffer occupancy threshold policy aimed at assuring service level agreement compliance for conforming (i.e., committed) traffic, and performance maximization for non-conforming (i.e., excess) traffic. Our analysis shows how the different system parameters: total buffer size, threshold position, and offered load control performance for the committed and excess traffic. Our results allow engineering of the system parameters aimed at assuring high service level agreement compliance for conforming (i.e., committed) traffic, and performance maximization for non-conforming (i.e., excess) traffic.

I. Introduction

In the ongoing work aimed at finding the way to transform the internet from the single class best effort service, to providing a variety of service classes offering different performance guarantees (QoS), simple coarse schemes and lightweight hardware support have become popular. Some such schemes are based on the concept of classification and performance level assignments at the edge of the networks. Packets are marked or tagged accordingly, and this marking is used to apply differentiated handling of the packets in the core of the network. These ideas took form in the extensive work of the differentiated services (DiffServ) working group of the IETF [NBBB98], [BBC+98], [NC01], [Gro02]. They were later also incorporated into the MPLS world in the form of MPLS DiffServ-TE technology [FWD+02], and recently introduced into the Metro Ethernet world, with the standardization efforts of the Metro Ethernet Forum [For04], [San04].

A typical contract between a customer and a provider is stated in terms of a service level agreement (SLA). In its simplest form it ensures the customer a minimum or expected bandwidth for its usage and may allow additional bandwidth to be used based on availability. The SLA may also define delay requirements (e.g., for real time applications).

We examine a typical case where several classes of services are defined. Customers requiring high performance (e.g., low delay and loss as defined in their SLAs) are assigned to the high priority class. Other customers are assigned to the lower priority classes with lower performance. The packets of a given class, that conform to the agreed expected bandwidth, are termed in this work committed bandwidth traffic of that class, and the packets that do not conform are termed excess traffic (these are sometimes termed ‘in‘ and ‘out‘ packets, respectively).

Typically at the ingress of the network, the provider monitors each class of traffic and marks the packets that exceed the committed rate as excess. The provider assures negligible drop probability for the committed traffic (of all classes) even during congestion periods. When congestion occurs the policy is to drop the excess traffic with higher probability. Specifically this policy means that in a congestion period it is preferable to drop excess traffic of high priority to dropping low priority committed traffic.

Implementation of such QoS policies in the network core nodes may be done by means of packet scheduling and buffer management mechanisms that handle packets according to their marked class and rate conformance. As mentioned, packet scheduling schemes set to achieve delay and loss differentiation may employ some kind of priority queueing. Buffer management in congestion periods typically includes a packet drop policy used to control and manage congestion.

Queue management has been studied extensively [KK80], [CH98] and complete memory sharing among all classes has been shown to provide optimal throughput / delay performance and maximal utilization of available memory in the system [FT89], [IKM01].

There is a long line of work that examines threshold policies for two (or more) types of packets that share a single FIFO buffer [CGK94], [CGGK95], [AMRR00], [LPS02], these deal with, either the case of a single class of packets some which are marked as discard eligible (e.g., non rate conforming), or the case of multiple classes of packets that are sharing a single FIFO buffer. In this paper we consider, for the first time, the case of multiple priority classes of packets each having two discard levels, namely committed and excess packets.

The system proposed in this work can be considered as a simple low cost and fast core node supporting coarse QoS differentiation. The system is based on a single shared memory space accommodating multiple FIFO queues (one per priority class). Packets are serviced according to a strict priority scheduling policy. A simple total-occupancy-threshold policy is used for buffer management (see Sec. 3.3 in [CGK94]).

We wish to analyze and study the behavior of such a system and provide guidelines for setting optimal system parameters (thresholds and buffer sizes) given traffic conditions. Our goal is to satisfy the requirements of the SLA defined for the committed traffic (i.e., negligible drop probability, and adequate delay for each priority class) while maximizing the utilization of available excess bandwidth to serve the revenue generating excess traffic. This is to be achieved with minimal memory requirements.

To this end we use the following model (see figure 1). The system is comprised of two priority queues:

- Priority queue 1 (high priority) serves two traffic types, committed and excess. The excess traffic is managed by means of a threshold, α_{1E}, which inhibits priority excess traffic acceptance based on total buffer space portion occupied (by all priorities and discard levels).
- Priority queue 0 (low priority) has two traffic types, committed and excess. The threshold α_{0E}, has the same meaning as that defined for priority 1 traffic.

Service is non preemptive.

Our goal is to present a simple tractable model to allow efficient analysis and calculation. For simplicity we first present analysis that uses a simpler model. In this simplified model the high priority queue is presented with both excess and committed traffic, and the low priority traffic is presented with committed traffic only. Thus we have three packet types: high priority committed, high priority excess, and low priority committed. Second, we use Poisson arrival processes to model all incoming traffic types. Third we deal with the finite nature of our queue in our model only to the extent needed to analyze committed traffic loss. For the most part we assume that the headroom (i.e., the buffer space above the threshold) is infinite. This assumption is based on two facts: 1. The marking process employed at the network ingress, controls the committed traffic rate and characteristics. 2. The system design process is aimed at avoiding committed traffic loss. Indeed we show that the system designed this way has a quickly dropping buffer-occupancy distribution function above the threshold. This allows for for the infinite headroom assumption given that the actual headroom allocated is large enough. Generalizations of the system, doing without the above mentioned simplifications, are addressed later in the work.

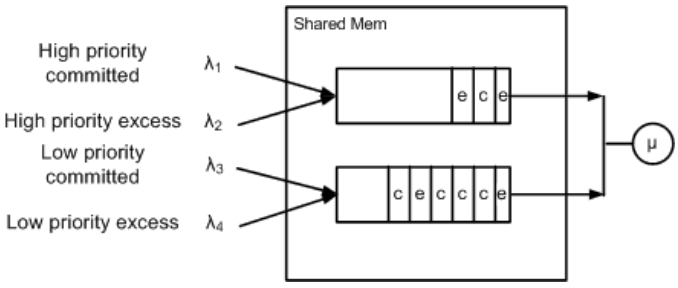

Fig. 1. System model

We start by giving exact numerical analysis of this and derive the loss and delay of the different type of packets. Next we present an approximated and simple analysis that allows us to easily explore the trade-off's of the system parameters. Next we present a simulation study of this system to validate our approximation assumptions.

??? Finally, we present a novel idea of a new type of threshold that we term *cross class threshold*. This threshold limits excess traffic on some class based on the length of the queue of lower priority committed traffic. This enables us better utilization of the buffer space and better absorption of bursts.

The rest of the paper is structured as follows.... In Section II-A we present an exact analysis of the system described above, and in Section II-B we show how it can be done efficiently. Section V presents simulation results and Section **??** presents the new cross class threshold.

II. The System Model

Two queues share a buffer space of n packets (or cells). The high priority queue serves committed traffic and excess traffic packet arrivals modeled by a Poisson

process of rates λ_1 and λ_2 respectively. The low priority queue serves committed traffic packet arrivals, also modeled as a Poisson process at rate λ_3. Service rate is μ (see figure 1). The threshold is denoted $n_{th} = \alpha_{1E} n$. When the total occupancy of the buffer is above this threshold, excess high priority traffic is rejected and lost.

A. Exact analysis

In this section we present an exact analysis of the system and derive for each class and priority the throughput and average delay.

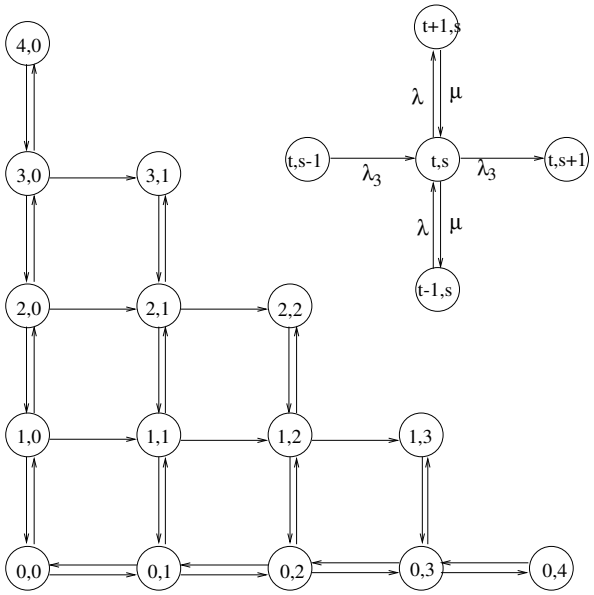

Fig. 2. A Markov chain for the two queue system.

The above system can be modeled by a continuous-time Markov chain with $(n+1)(n+2)/2$ states as illustrated in figure 2 for the case where $n = 4$. Each state is represented by the ordered pair (t, s), where t is the number of high priority packets in the buffer and s the number of low priority packets. The infinitesimal transition rates from state (t, s) to state (t', s'), $q_{t,s,t',s'}$ are (see Figure 2)

$$
\begin{aligned}
q_{t,s,t-1,s} &= \mu \\
q_{0,s,0,s-1} &= \mu \\
q_{t,s,t,s+1} &= \lambda_3 \\
q_{t,s,t+1,s} &= \begin{cases} \lambda_1 + \lambda_2 & \text{if } t+s \le n_{th} \\ \lambda_1 & \text{if } t+s > n_{th} \end{cases} \\
-q_{t,s,t,s} &= \begin{cases} \lambda_1 + \lambda_2 + \lambda_3 & \text{if } t+s = 0 \\ \lambda_1 + \lambda_2 + \lambda_3 + \mu & \text{if } 0 < t+s \le n_{th} \\ \lambda_1 + \lambda_3 + \mu & \text{if } t+s > n_{th} \end{cases}
\end{aligned}
\tag{1}
$$

Note that $-q_{t,s,t,s}$ is the transition rate out of state (t, s).

To find the steady state probabilities, $\pi_{t,s}$, we can solve the the system equilibrium equations, $\vec{\pi} Q = 0$ (Q is derived directly from the infinitesimal transition rates, $\vec{\pi}$ is the vector of steady state probabilities), together with the probability conservation relation, $\sum_{(t,s)} \pi_{t,s} = 1$. This numerical solution requires $O(n^{2(2+\alpha)})$ basic operations, where $O(x^{2+\alpha})$ is the number of operations used by the matrix inversion algorithm for an $x \times x$ matrix (for the best known matrix inversion algorithm $\alpha > 0.5$). I.e., the solution requires $O(n^5)$ operations. In the following, we shall describe methods to make the problem more tractable, by presenting a recursive solution that requires only $O(n^3)$ operations.

We are interested in the regime where $\lambda_1 + \lambda_3$ is smaller than μ or else committed traffic will discarded with probability 1. Under this condition we can calculate the drop probability η_i as follows:

$$\eta_1 = \sum_{i=0}^{n} \pi_{i,n-i} \tag{2}$$

$$\eta_2 = \sum_{i+j>n_{th}} \pi_{i,j} \tag{3}$$

$$\eta_3 = \sum_{i=0}^{n} \pi_{i,n-i} \tag{4}$$

Note that by definition $\eta_1 = \eta_3$ which shows that there is no preference between the two priority classes in the acceptance probability.

To estimate the average delay for the lower class: Let $\bar{N}_i$ be the average number of cells of type i in the system.

$$\bar{N_3} = \sum_{i,j} \pi_{i,j} j$$

Using Little's Law we know that the average delay, T_3, is given by

$$T_3 = \frac{\bar{N_3}}{(1-\eta_3)\lambda_3} = \frac{\sum_{i,j} \pi_{i,j} j}{(1 - \sum_{i=0}^{n} \pi_{i,n-i})\lambda_3} \tag{5}$$

B. Reducing the Analysis Complexity Using Recurrence

We can reduce the computation complexity above by using recursion. Our aim is to write the steady state probabilities of all the system states, $\pi_{t,s}$, as functions of $\pi_{0,s}$, $0 \le s \le n$. Then, we can write n equilibrium equations and together with the probability conservation equation we obtain $n+1$ linear equations that can be solved with complexity of $O(n^3)$.

Using the Markov chain illustrated in figure 2 and the transition rates of Eq. 13 (also illustrated in figure 2), we

can write the following $n(n+1)/2$ equilibrium equations

$$\begin{aligned}
-q_{t,s,t,s}\pi_{t,s} &= q_{t,s-1,t,s}\pi_{t,s-1} \\
&\quad +q_{t+1,s,t,s}\pi_{t+1,s} + q_{t-1,s,t,s}\pi_{t-1,s} \quad (6)\\
-q_{t,0,t,0}\pi_{t,0} &= q_{t+1,0,t,0}\pi_{t+1,0} + q_{t-1,0,t,0}\pi_{t-1,0} \\
&\quad 1 \le t \le n-1 \\
-q_{0,s,0,s}\pi_{0,s} &= q_{0,s-1,0,s}\pi_{0,s-1} + q_{1,s,0,s}\pi_{1,s} \\
&\quad +q_{0,s+1,0,s}\pi_{0,s+1} \qquad 1 \le s \le n-1 \\
-q_{0,0,0,0}\pi_{0,0} &= q_{1,0,0,0}\pi_{1,0} + q_{0,1,0,0}\pi_{0,1}
\end{aligned}$$

Substituting the values for $q_{t,s,t',s'}$ from Equation 13 in Equation 6 we can rewrite the equilibrium equations as

$$\begin{aligned}
-(\lambda_1+\lambda_2+\lambda_3+\mu)\pi_{t,s} &= \lambda_3\pi_{t,s-1} + \mu\pi_{t+1,s} \\
&\quad +(\lambda_1+\lambda_2)\pi_{t-1,s}, \\
&\quad t>0, t+s \le n_{th} \\
-(\lambda_1+\lambda_3+\mu)\pi_{t,s} &= \lambda_3\pi_{t,s-1} + \mu\pi_{t+1,s} \\
&\quad +(\lambda_1+\lambda_2)\pi_{t-1,s}, \\
&\quad t>0, t+s = n_{th}+1 \\
-(\lambda_1+\lambda_3+\mu)\pi_{t,s} &= \lambda_3\pi_{t,s-1} \\
&\quad +\mu\pi_{t+1,s} + \lambda_1\pi_{t-1,s}, \\
&\quad t>0, t+s > n_{th}+1 \\
-(\lambda_1+\lambda_2+\lambda_3+\mu)\pi_{t,0} &= \mu\pi_{t+1,0} + (\lambda_1+\lambda_2)\pi_{t-1,0}, \\
&\quad 1 < t \le n_{th} \\
-(\lambda_1+\lambda_3+\mu)\pi_{n_{th}+1,0} &= \mu\pi_{n_{th}+2,0} + (\lambda_1+\lambda_2)\pi_{n_{th},0} \\
-(\lambda_1+\lambda_3+\mu)\pi_{t,0} &= \mu\pi_{t+1,0} + \lambda_1\pi_{t-1,0}, \\
&\quad n_{th}+1 < t \le n-1 \\
-(\lambda_1+\lambda_2+\lambda_3+\mu)\pi_{0,s} &= \lambda_3\pi_{0,s-1} + \mu\pi_{1,s} + \mu\pi_{0,s+1}, \\
&\quad s \le n_{th} \\
-(\lambda_1+\lambda_3+\mu)\pi_{0,s} &= \lambda_3\pi_{0,s-1} + \mu\pi_{1,s} + \mu\pi_{0,s+1}, \\
&\quad s > n_{th} \\
-(\lambda_1+\lambda_2+\lambda_3)\pi_{0,0} &= \mu\pi_{1,0} + \mu\pi_{0,1}
\end{aligned} \quad (7)$$

Now, we can write the following recursion relations for $\pi_{t,s}$, $t>0$:

$$\begin{aligned}
\pi_{1,0} &= (-q_{0,0,0,0}\pi_{0,0} - q_{0,1,0,0}\pi_{0,1})/q_{1,0,0,0} \quad (8)\\
\pi_{1,s} &= (-q_{0,s,0,s}\pi_{0,s} - q_{0,s-1,0,s}\pi_{0,s-1} - q_{0,s+1,0,s}\pi_{0,s+1}) \\
&\quad /q_{1,s,0,s} \qquad s = 1,2,\ldots,n-1 \\
\pi_{t,0} &= (-q_{t-1,0,t-1,0}\pi_{t-1,0} - q_{t-2,0,t-1,0}\pi_{t-2,0}) \\
&\quad /q_{t,s,t-1,s} \quad 2 \le t \le n \\
\pi_{t,s} &= (-q_{t-1,s,t-1,s}\pi_{t-1,s} - q_{t-2,s,t-1,s}\pi_{t-2,s} - \\
&\quad q_{t-1,s-1,t-1,s}\pi_{t-1,s-1})/q_{t,s,t-1,s} \\
&\quad t = 2,3,\ldots,n \quad s = 1,2,\ldots,n-t
\end{aligned}$$

The above recurrence suggests that all $\pi_{t,s}$ can be written as functions of $\pi_{0,s}$, i.e.,

$$\pi_{t,s} = \sum_{l=0}^{n} C_{t,s}(l)\pi_{0,l}, \quad (9)$$

It is easier to calculate the recurrence for the coefficients, $C_{t,s}(l)$, rather than directly for $\pi_{t,s}$. First, we calculate the coefficients of $\pi_{1,s}$ by

$$\begin{aligned}
C_{1,s}(s) &= -q_{0,s,0,s}/q_{1,s,0,s} \quad s = 0,1,2,\ldots,n-1 \\
C_{1,s}(s-1) &= -q_{0,s-1,0,s}/q_{1,s,0,s} \quad s = 1,2,\ldots,n-1 \\
C_{1,s}(s+1) &= -q_{0,s+1,0,s}/q_{1,s,0,s} \quad s = 0,1,2,\ldots,n-1 \\
C_{1,s}(l) &= 0 \quad |l-s| > 1
\end{aligned} \quad (10)$$

Next, we calculate the coefficients of $\pi_{t,s}$ for $t = 2,3,\ldots,n-1$:

$$\begin{aligned}
C_{t,s}(m) &= (q_{t-1,s,t-1,s}C_{t-1,s}(m) \\
&\quad -q_{t-1,s-1,t-1,s}C_{t-1,s-1}(m) \\
&\quad -q_{t-2,s,t-1,s}C_{t-2,s}(m))/q_{t,s,t-1,s}
\end{aligned} \quad (11)$$

The recurrence calculation requires $O(n^3)$ operations. $n+1$ equilibrium equations are not used to derive the recurrence, thus n of them can be used together with the probability conservation equation, in equation system 12, to achieve the following $n+1$ linear equation system, whose solution complexity is lower than $O(n^3)$.

$$\begin{aligned}
-q_{t,n-t,t,n-t}\pi_{t,n-t} &= q_{t,n-(t+1),t,n-t}\pi_{t,n-(t+1)} \quad (12)\\
&\quad q_{t-1,n-t,t,n-t}\pi_{t-1,n-t} \\
&\quad 1 \le t \le n-1 \\
-q_{0,n,0,n}\pi_{0,n} &= q_{0,n-1,0,n}\pi_{0,n-1} \\
\sum_{(t,s)} \pi_{t,s} &= 1
\end{aligned}$$

Using the recurrence on the coefficients we can write Eq. 12 as

$$\begin{aligned}
&-\sum_{m=0}^{n} q_{t,n-t,t,n-t}C_{t,n-t}(m)\pi_{0,m} = \\
&\quad \sum_{m=0}^{n} (q_{t,n-(t+1),t,n-t}C_{t,n-(t+1)}(m) \\
&\quad +q_{t-1,n-t,t,n-t}C_{t-1,n-t}(m))\pi_{0,m} \\
&\quad 1 \le t \le n-1
\end{aligned}$$

and rewrite the probability conservation equation as

$$\sum_{t=0}^{n-1}\sum_{s=0}^{n-t}\sum_{m=0}^{n} C_{t,s}(m)\pi_{0,m} = 1$$

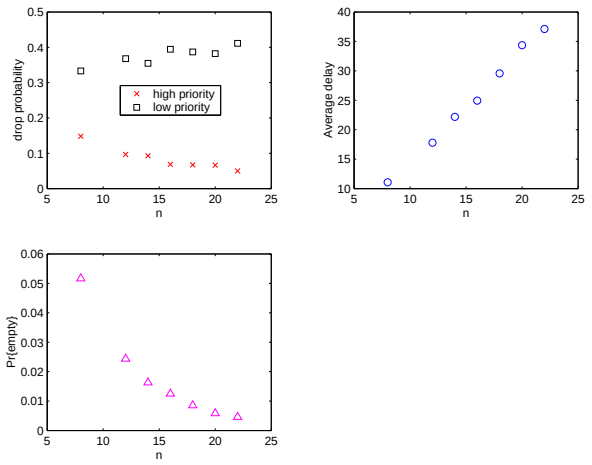

Fig. 3. The analysis as a function of the buffer size, n. Results are for $\lambda_1 = \lambda_2 = \lambda_3 = 0.4$ and $\alpha_{1E} = 0.8$

C. Numerical results

Figure 3 shows that for fairly small values of n the analysis already reaches steady state in the drop probability. As can be expected the delay of class 0 packets grows with the increase in the number of buffers, surprisingly it seems to grow linearly.

III. Approximated Analysis

The complexity of the numerical solution presented above may still be too high to allow solving the system for buffer sizes exceeding a few tens of packets. For the case where system behavior is controlled by total occupancy thresholds we suggest instead to model the system by a single parameter, its total occupancy, as explained below.

A. Analysis of total system occupancy

We suggest looking at the one dimensional state space representing the total occupancy of the shared memory buffer, namely, the number of packets (of all types) present in the system at a given moment.

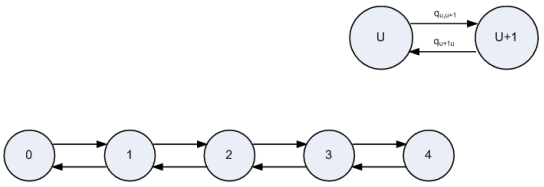

Fig. 4. A Markov chain model for the total system occupancy

In this analysis the system can be modeled by a continuous-time Markov chain with $n+1$ states as illustrated in figure 4 for the case of n=4.

The state transition probabilities are given by

$$q_{u,u+1} = \begin{cases} \lambda_1+\lambda_2+\lambda_3 & \text{if } u \le n_{th} \\ \lambda_1+\lambda_3 & \text{if } n_{th} < u < n \\ 0 & \text{if } u = n \end{cases} \quad (13)$$

$$q_{u,u-1} = \begin{cases} \mu & \text{if } 0 < u \le n \\ 0 & \text{if } u = 0 \end{cases}$$

To find the steady state probabilities, π_u, we solve the system equilibrium equations, together with the probability conservation relation:

$$q_{u,u+1}\pi_u = q_{u,u-1}\pi_{u+1} \qquad 0 \le u \le n-1 \quad (14)$$

$$\sum_{u=0}^{n} \pi_u = 1 \quad (15)$$

we define ρ_u for the unrestricted full load case, and ρ_r for the restricted load case (when occupancy is over the threshold):

$$\begin{aligned} \rho_u &= (\lambda_1+\lambda_2+\lambda_3)/\mu \quad (16) \\ \rho_r &= (\lambda_1+\lambda_3)/\mu \end{aligned}$$

solving this equation explicitly yields:

$$\pi_u = \begin{cases} \pi_0(\rho_u)^u & \text{if } u \le n_{th}+1 \\ \pi_{(n_{th}+1)}(\rho_r)^{(u-n_{th}-1)} & \text{if } n_{th}+1 < u \le n \end{cases} \quad (17)$$

and applying the probability conservation (eq. 15) yields:

$$\pi_0 = \frac{(1-\rho_u)(1-\rho_r)}{(1-\rho_r)(1-\rho_u^{n_{th}+2}) + \rho_u^{n_{th}+1}\rho_r(1-\rho_u)(1-\rho_r^{n-n_{th}-1})} \quad (18)$$

The drop probabilities for the different traffic types (i.e., high priority committed traffic, high priority excess traffic, and low priority committed traffic, denoted by η_1, η_2, and η_3 respectively) can be calculated as follows:

$$\begin{aligned} \eta_1 &= \pi_n \\ \eta_2 &= \sum_{u=n_{th}+1}^{n} \pi_u \quad (19) \\ \eta_3 &= \pi_n \end{aligned}$$

explicitly:

$$\begin{aligned} \eta_1 &= \pi_0 \rho_u^{n_{th}+1} \rho_r^{n-n_{th}-1} \\ \eta_2 &= \pi_0 \rho_u^{n_{th}+1} \frac{1-\rho_r^{n-n_{th}}}{1-\rho_r} \quad (20) \\ \eta_3 &= \eta_1 \end{aligned}$$

We are interested in the regime where $(\lambda_1+\lambda_3)/\mu < 1$ or else committed traffic will be lost with probability 1. Furthermore we look mainly at the case when the total load, $(\lambda_1 + \lambda_2 + \lambda_3)/\mu$, exceeds unity as it represents periods of congestion.

Figure 5 shows total occupancy distribution for two loads points of 1.05 and 1.15 (unless otherwise specified the system load is defined as the aggregate load of all packet types. Also unless otherwise specified the rate is equal for all types. i.e., $\lambda_1 = \lambda_2 = \lambda_3$). This figure demonstrates that in the cases of interest the total occupancy probability distribution drops fast above the threshold. This allows the infinite buffer assumption, given that the actual space above the threshold is large enough.

Figure 6 shows the acceptance of the various packet types as a function of the threshold value at the same two load values (committed traffic in these figures is not lost according to our infinite capacity assumption). Both figures include the simulation results for comparison see section V. See section IV for more details.

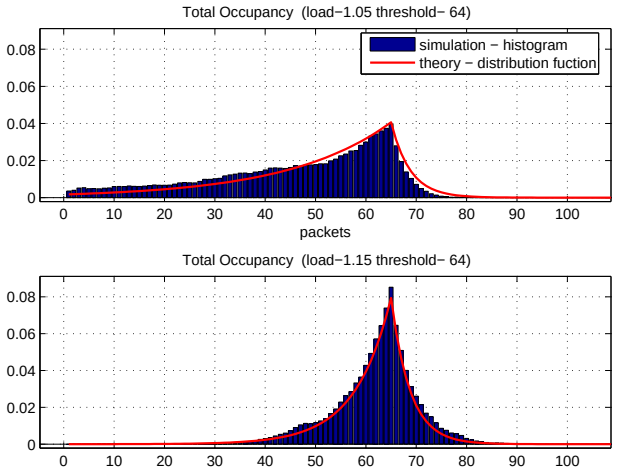

Fig. 5. Total system occupancy distribution function at two load points

B. Delay analysis

We present here an analysis due to [Kle76] concerning multi-priority infinite capacity queueing. The following section adapts a similar method for the approximate analysis of our model. Let a 'tagged' customer arrive at the system. The 'tagged' customer's total waiting time (in queue) is comprised of three parts. The first part is due to the customer found in service (the system is nonpreemptive). The second part is due to customers of equal or greater priority already present in the queue and

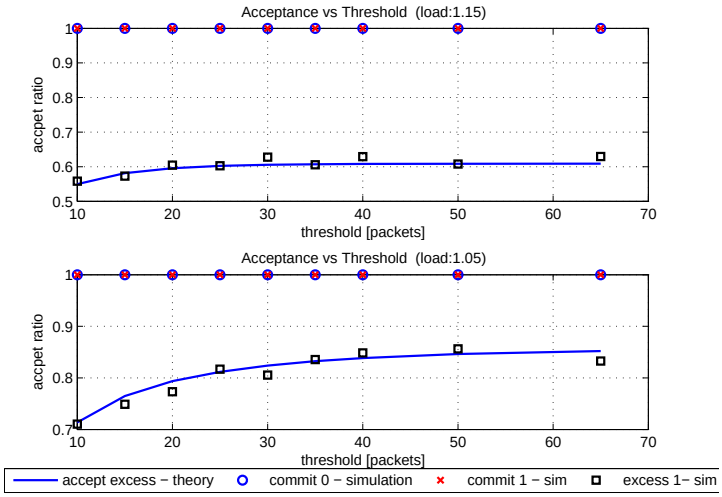

Fig. 6. Acceptance Vs. Threshold at two load points. The buffer size is large enough so committed traffic loss probability is negligable

subscript p	Priority index. Higher values mean higher priorities ($p \in \{0, 1, ..., P\}$).
W_p	Average waiting time in queue for a customer of priority p.
$\hat{W_0}$	Average time from the random instant of customer arrival until the completion of the current service
N_{ip}	number of customers from priority class i found in queue by an arriving tagged customer of priority p and receiving service before him.
M_{ip}	Number of customers from group i who arrive while the tagged customer is in queue, and receive service before him.
$\bar{x_p}$	Average service time for customers of priority p
$\hat{\lambda_p}$	Average arrival rate of priority p customers

TABLE I

SYMBOLS USED IN MULTIPLE PRIORITY DELAY CALCULATIONS

receiving service before the tagged customer. The Third part is due to customers of higher priority arriving while the tagged customer is waiting in the queue and receiving service before him. We use the definitions of Table I. Please note the carets in the notations: $\hat{\lambda_p}$ represent the average input rate into the class p queue (as opposed to λ_1,λ_2 and λ_3 representing the poisson process rate of the three traffic types). And $\hat{W_0}$ is the random residual service time as opposed to the waiting time of class p denoted W_p. And so:

$$W_p = \hat{W_0} + \sum_{i=0}^{P} \bar{x_i}(N_{ip} + M_{ip}) \qquad (21)$$

By our queueing discipline:

$$N_{ip} = 0 \text{ if } i = 0, 1, 2, ..., p-1$$
$$M_{ip} = 0 \text{ if } i = 0, 1, 2, ..., p$$

(22)

By little's theorem:

$$\begin{aligned} N_{ip} &= \hat{\lambda_i} W_i \text{ if } i = p, p+1, ..., P \\ M_{ip} &= \hat{\lambda_i} W_p \text{ if } i = p+1, p+2, ..., P \end{aligned} \tag{23}$$

This system can be solved recursively to obtain:

$$\begin{aligned} W_p &= \frac{\hat{W_0}}{(1-\sigma_p)(1-\sigma_{p+1})} \\ \sigma_p &= \sum_{i=p}^{P} \rho_i \\ \rho_i &= \bar{x_i}\hat{\lambda_i} \end{aligned} \tag{24}$$

$\hat{W_0}$ is due to residual theory (see [Kle76] Sec 1.7).

$$\hat{W_0} = \sum_{i=0}^{P} \rho_i \frac{\bar{x_i^2}}{2\bar{x_i}} \tag{25}$$

Which is the sum of the average residual service times for each priority class, weighted by the chance of one of its members being in service (ρ_i).

C. Adaptation to the model

For our approximated analysis of the delay in the threshold governed priority queue we will use a similar steady state approach to the one described in [Kle76]. Here again we make the infinite capacity approximation (see section I).

We now claim that under the above assumptions we can approximate the average waiting time of the high priority queue in our system using the results from the infinite case presented in section **??**.

Let us write the waiting time equation for priority 1 (see equation 21 above) for our case:

$$W_1 = \hat{W_0} + \bar{x_1} N_1$$

substituting $\hat{\lambda_1} W_1$ for N_1 by little's theorem and rearranging yields:

$$W_1 = \frac{\hat{W_0}}{1 - \bar{x_1}\hat{\lambda_1}} \tag{26}$$

We adapt this result to the model under consideration by recalculating $\hat{\lambda_i}$ and $\hat{W_0}$ using the steady state distribution of the total system occupancy obtained above (see section III-A). In our case (exponential service at rate μ for all priorities) both $\bar{x_i}$ and $\frac{\bar{x_i^2}}{2\bar{x_i}}$ reduce to $1/\mu$ for every i. The chance of the server being free is $1 - \pi_0$ (see equation 18). Therefore $\hat{W_0}$ (see equation 25) can be written as:

$$\hat{W_0} = \frac{1}{\mu}(1 - \pi_0) \tag{27}$$

Since we consider infinite total capacity, the total system occupancy distribution function (equation 17) becomes:

$$\begin{aligned} \pi_u &= \pi_0(\rho_u)^u \quad \text{if } u \leq n_{th} + 1 \\ \pi_u &= \pi_{n_{th}}(\rho_r)^{(u - n_{th})} \quad \text{if } n_{th} + 1 < u \end{aligned} \tag{28}$$

where

$$\pi_0 = \frac{(1-\rho_u)(1-\rho_r)}{(1-\rho_r)(1-\rho_u^{n_{th}+2}) + \rho_u^{n_{th}+1}\rho_r(1-\rho_u)} \tag{29}$$

The average input rate for the high priority queue can now be calculated as:

$$\begin{aligned} \hat{\lambda_1} &= (\lambda_1 + \lambda_2) \cdot P(occupancy \leq n_{th}) \\ &\quad + (\lambda_1) \cdot P(n_{th} < occupancy) \end{aligned} \tag{30}$$

yielding :

$$\hat{\lambda_1} = \pi_0[(\frac{1-\rho_u^{n_{th}+1}}{1-\rho_u})(\lambda_1 + \lambda_2) + \rho_u^{n_{th}+1}(\frac{1}{1-\rho_r})(\lambda_1)] \tag{31}$$

and so we have:

$$W_1 = \frac{\hat{W_0}}{1 - \frac{1}{\mu}\hat{\lambda_1}} \tag{32}$$

The full result is obtained by combining equations 32, 27, 29 and 31.

Using this result together with Little's theorem we can calculate the waiting time of the low priority queue. The average occupancy of the low priority queue is:

$$N_0 = \hat{\lambda_0} W_0 = N - N_1 \tag{33}$$

$N_1 = W_1\hat{\lambda_1}$, and N can be calculated using the results of the total system occupancy distribution (eq. 28), yielding:

$$\begin{aligned} N &= \frac{\pi_0}{(1-\rho_u)^2}[\rho_u^{n_{th}+2}((n_{th}+1)\rho_u - (n_{th}+2)) + \rho_u] - \\ &\quad \frac{\pi_0}{(1-\rho_r)^2}[\rho_r\rho_u^{n_{th}+1}((n_{th}+1)\rho_r - (n_{th}+2))] \end{aligned} \tag{34}$$

$\hat{\lambda_0}$ in this case is equivalent to λ_3.

and so:

$$W_0 = \frac{N - W_1\hat{\lambda_1}}{\lambda_3} \tag{35}$$

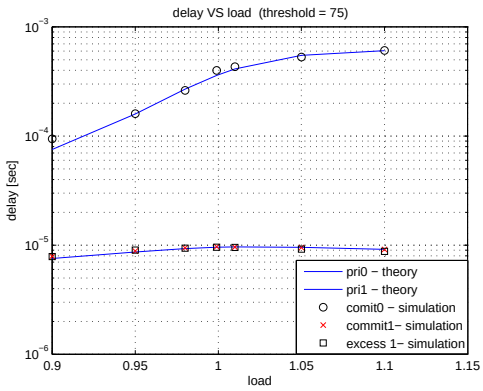

Fig. 7. Delay vs. Load

In figure 7 the expected delay is shown (theoretical calculations and corresponding simulated results) as a function of the aggregate load for all priorities. The figure shows that our analysis agrees with the simulation we conducted. The simulation model is discussed in section V. For a full description of system behavior see section IV.

IV. System behavior and trad-offs

Using the above results we can now study the system behavior and trade-offs presented as a function of load and threshold selection. We are interested in the regime where the aggregate input rate is close to the service rate (i.e., time of congestion). We assume that the committed traffic is allocated enough resources to keep loss low (namely, committed aggregate input rate, $\lambda_1 + \lambda_3$, is lower than the service rate, and buffer space above the threshold ,i.e., $n - n_t h$, is allocated). This is a logical common policy. Low loss can be verified by checking that expected committed traffic loss ratios (η_1 or equivalently η_3 of equation 19) are negligible.

In figures 8, 9 and 10 we show the effect of different loads and threshold values on the service level received by the committed and excess traffic of both priorities. Looking first at high priority (committed and excess) traffic delay (figure 8) we observe that the strict priority service scheme promises low delay that is affected by the threshold only when it becomes too low. This phenomenon is due to higher rejection of excess traffic as the threshold is lowered. This reduces the total load on the high priority queue and thus lowers the average delay of high priority packets. On the other hand (see figure 9), low priority traffic suffers a delay that grows linearly with the threshold value (this is the case in the regime of interest: where the aggregate load is higher than unity). This is understood since the total queue length is controlled by the threshold in congestion periods.

Finally looking at the excess traffic acceptance, we see that for each load value there is a maximum acceptance ratio that can be reached by raising the threshold. This maximum ratio represents full utilization of service bandwidth left over after all committed traffic is served. It can be seen that for higher load values low threshold values suffice to reach the maximum utilization. This property of the system can also be seen in the excess rejection depicted in figure 11. This behavior is due to the fact that at lower loads the server's idle probability (π_0) is significant (see figure 5). Increasing the effective queue length for the excess traffic (raising the threshold), lowers this probability and **increases the utilization of the server**, resulting in more excess traffic throughput.

To summarize: a reasonable design procedure would be to set committed traffic bandwidth share and loss probability targets (these would be in compliance with the various SLA commitments to the customers sharing this link, and monitored by a marking scheme). These performance targets for the committed traffic can be achieved by allocating sufficient headroom above the threshold so that even at high congestion periods loss probability remains low. Next the value of the threshold (and thus the total memory space allocated to the queue) can be set. The threshold selected is set to achieve the desired tradeoff between excess traffic acceptance and low priority delay. In addition the above analysis shows that for a given expected maximum aggregate link load, there is a threshold value region above which raising the threshold does not significantly improve acceptance ratio for excess traffic.

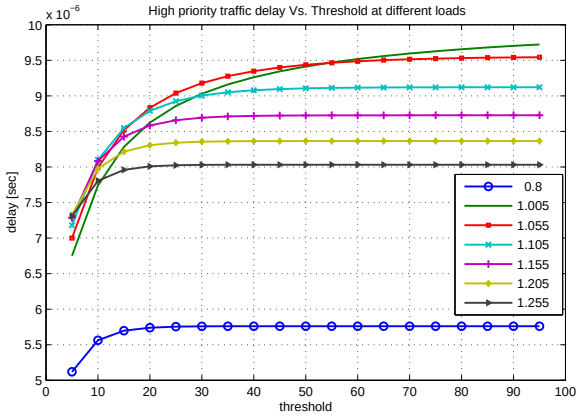

Fig. 8. High priority delay

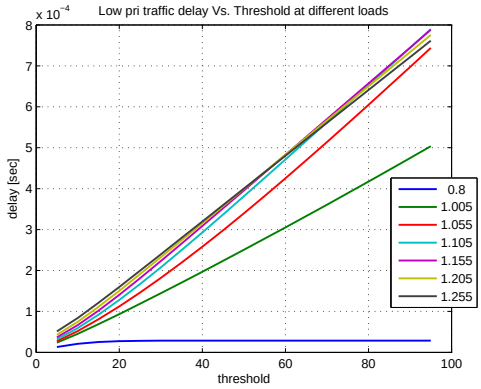

Fig. 9. Low priority delay

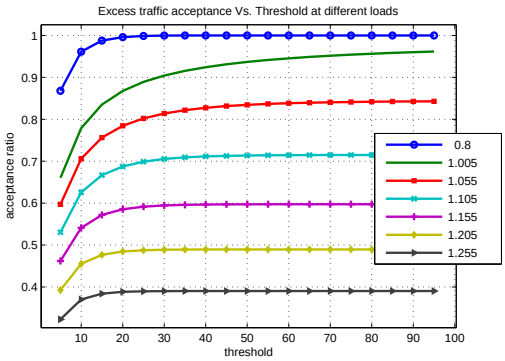

Fig. 10. Excess acceptance

V. SYSTEM SIMULATION

The computer simulation model used throughout this paper has the following characteristics: The packet arrival process is a Poisson process that is an aggregate of three independent processes (one for each packet type). Unless otherwise stated the rates of these three processes are equal (i.e., $\lambda_1 = \lambda_2 = \lambda_3$). The packet lengths are exponentially distributed and thus service time is also exponentially distributed.

Results throughout the paper (specifically those base on approximations) are compared to results of this computer simulated model and seem to confirm their validity .

VI. BURSTY TRAFFIC SIMULATION

So far we have assumed a Poisson arrival process. The main reason for this assumption is mathematical tractability. To make our study more complete, we now extend our model to include bursty traffic conditions. We do this by means of computer simulation to obtain some qualitative results. For the simulation of bursty traffic we use an on/off source, where the on and off

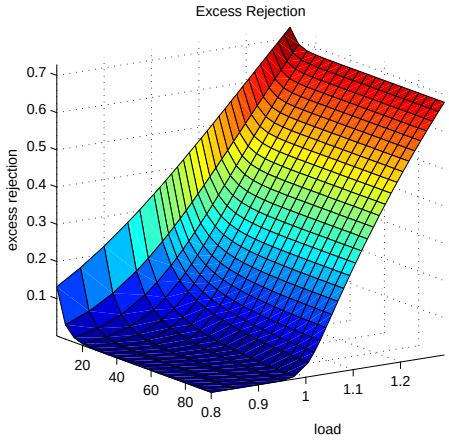

Fig. 11. Excess traffic rejection

periods are exponentially distributed (Markov Modulated Poisson Process - MMPP with two states).

In the following simulation run results (figures 12 and 13), each packet type source, was modulated by a two state (on/off) Markov chain. Steady state probability for the 'ON' state is 0.5, and transition rate (on $\rightarrow$ off) is $\lambda_i/10$.

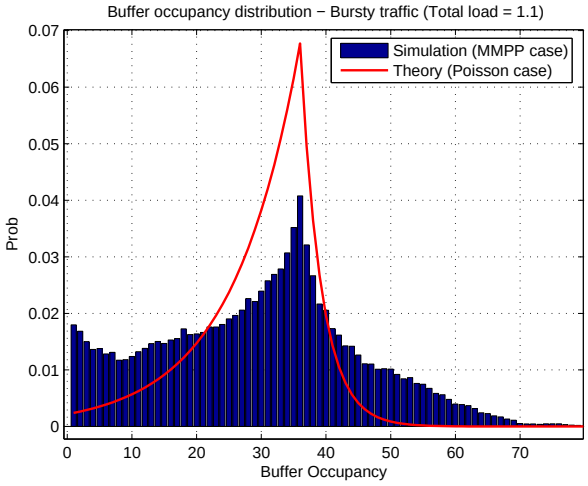

Fig. 12. Occupancy distribution histogram with Markov modulated Poisson arrival processes - Simulation of bursty traffic. Theory graph (line) is the corresponding distribution function in the Poisson arrival process case given for comparison

Three phenomena can be seen. First, acceptance of excess traffic is more affected by threshold values than traffic of the same load modeled by a Poisson arrival process. This can be attributed to the fact that in the bursty traffic case the probability of an idle server increases (figure 13), similarly to the cases of lower load points in the Poisson arrival model (see section IV). If the threshold occupancy value is selected high

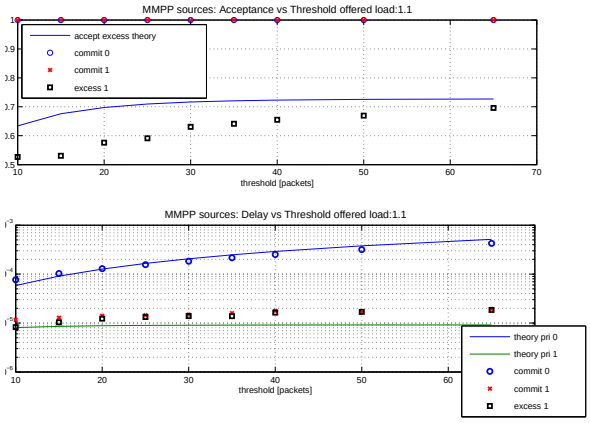

Fig. 13. Delay and acceptance with Markov modulated Poisson arrival processes - Simulation of bursty traffic behavior of the model. Theory graphs (lines) are the plain Poisson process results for comparison

enough so that the server is fully utilized the excess traffic throughput reaches its theoretical limit as in the Poisson arrival process case. Second we can see that burstiness has little effect on the delay of the low priority traffic (figure 12 bottom graph). This is can be explained by the averaging effect of the fully occupied low priority queue, making threshold value the dominant factor in low priority delay. Third, we see that high priority delay is affected by the threshold, increasing as the threshold value is raised and reaching a value more than twice as high as that of the Poisson case. As in the lower loads of the Poisson case (see section IV), the increasing delay is attributed to the increasing acceptance of the excess traffic (figure 12 top graph). The higher asymptotic delay is due to the fact that the bursts are of lengths comparable and even larger than the average queue length of the high priority queue (e.g., in the case shown in figures 13 and 12 the average burst size is ten packets, and the average high priority queue length in the corresponding Poisson case is 2.7 packets). Because of this situation the burstiness is affecting the average size of the high priority queue, increasing the its average delay.

VII. Excess low priority traffic

The approximated method shown above can be easily extended to get results for the more realistic model consisting of excess traffic of low priority as well (as described in the Introduction: section I).

We thus add a fourth arrival process of low priority excess traffic of rate λ_4. This traffic type will be controlled in a similar manner: when the portion of the buffer occupied, by all packet types, exceeds $\alpha_{0E} \leq \alpha_{1E}$, excess low priority traffic will be rejected and lost. This creates a policy where low priority excess traffic will be the first to be dropped when congestion occurs. If the congestion is not relieved and the buffer's occupancy continues to climb then high priority excess traffic will also be dropped. The main results for this extended model follow.

The new threshold will be denoted by $n_{th1} = \alpha_{0E} n$ and $n_{th2} = \alpha_{1E} n$ will be used in this section to denote the threshold controlling the high priority excess traffic. Also, for this section, we redefine the aggregate loads:

$$\begin{aligned} \rho_u &= (\lambda_1 + \lambda_2 + \lambda_3 + \lambda 4)/\mu \\ \rho_r &= (\lambda_1 + \lambda_2 + \lambda_3)/\mu \\ \rho_m &= (\lambda_1 + \lambda_3)/\mu \end{aligned} \tag{36}$$

The state probability function of the total buffer occupancy:

$$\pi_u = \begin{cases} \pi_0 \rho_u^u & \text{if } u \leq n_{th1}+1 \\ \pi_{(n_{th1}+1)} \rho_r^{(u-n_{th1}-1)} & \text{if } n_{th1}+1 < u \leq n_{th2}+1 \\ \pi_{(n_{th1}+1)} \rho_r^{(n_{th2}-n_{th1})} \rho_m^{(u-n_{th2}-1)} & \\ & \text{if } n_{th2}+1 < u \leq n \end{cases} \tag{37}$$

π_0 follows from the probability conservation (eq. 15):

$$\begin{aligned} \pi_0^{-1} &= \frac{{\rho_u}^{n_{th1}+2} - 1}{\rho_u - 1} + \frac{{\rho_u}^{n_{th1}+1}({\rho_r}^{n_{th2}-n_{th1}+1} - \rho_r)}{\rho_r - 1} \\ &\quad + \frac{{\rho_u}^{n_{th1}+1} {\rho_r}^{n_{th2}-n_{th1}} ({\rho_m}^{n-n_{th2}} - \rho_m)}{\rho_m - 1} \end{aligned} \tag{38}$$

Drop probabilities for the high priority committed traffic (denoted by η_1), high priority excess traffic (η_2), low priority committed traffic (η_3), and low priority excess traffic (η_4) are:

$$\begin{aligned} \eta_1 &= \pi_0 {\rho_u}^{n_{th1}+1} {\rho_r}^{n_{th2}-n_{th1}} {\rho_m}^{n-n_{th2}-1} \\ \eta_2 &= \pi_0 \frac{{\rho_u}^{n_{th1}+1} {\rho_r}^{n_{th2}-n_{th1}} (1 - {\rho_m}^{n-n_{th2}})}{1-\rho_m} \\ \eta_3 &= \eta_1 \end{aligned} \tag{39}$$

$$\eta_4 = \eta_2 + \pi_0 \frac{{\rho_u}^{n_{th1}+1} (1 - {\rho_r}^{n_{th2}-n_{th1}})}{1-\rho_r} \tag{40}$$

The average high priority input rate is now:

$$\begin{aligned} \hat{\lambda_1} &= (\lambda_1 + \lambda_2) \cdot P(occupancy \leq n_{th2}) \\ &\quad + (\lambda_1) \cdot P(n_{th2} < occupancy < n) \end{aligned} \tag{41}$$

yielding :

$$\begin{aligned} \hat{\lambda_1} &= \pi_0 [(\frac{{\rho_u}^{n_{th1}+1} - 1}{\rho_u - 1} + \frac{{\rho_u}^{n_{th1}+1} ({\rho_r}^{n_{th2}-n_{th1}} - 1)}{\rho_r - 1})(\lambda_1 + \lambda_2) \\ &\quad + (\frac{{\rho_u}^{n_{th1}+1} {\rho_r}^{n_{th2}-n_{th1}} ({\rho_m}^{n-n_{th2}-1} - 1)}{\rho_m - 1})(\lambda_1)] \end{aligned} \tag{42}$$

Having calculated these quantities we can now use equations 27 and 32 with the new $\hat{\lambda_1}$ and π_0 values, to get the expected delay for the high priority traffic, W_1.

To complete the analysis for this case, equation 35 will be rewritten as:

$$W_0 = \frac{N - W_1\hat{\lambda_1}}{\hat{\lambda_0}} \quad (43)$$

Where

$$\begin{aligned}\hat{\lambda_0} &= (\lambda_3+\lambda_4)\cdot P(occupancy \le n_{th1}) \\ &\quad +(\lambda_3)\cdot P(n_{th1} < occupancy < n) \end{aligned} \quad (44)$$

$$\begin{aligned}\hat{\lambda_0} &= \pi_0[(\frac{{\rho_u}^{n_{th1}+1}-1}{\rho_u-1})(\lambda_3+\lambda_4) \\ &\quad +(\frac{{\rho_u}^{n_{th1}+1}\left({\rho_r}^{n_{th2}-n_{th1}}-1\right)}{\rho_r-1} \\ &\quad +\frac{{\rho_u}^{n_{th1}+1}{\rho_r}^{n_{th2}-n_{th1}}\left({\rho_m}^{n-n_{th2}-1}-1\right)}{\rho_m-1})(\lambda_3)]\end{aligned} \quad (45)$$

The average total occupancy, N, is calculated using the total occupancy distribution function (equation 37):

$$N = \sum_{u=0}^{n} u\pi_u$$

$$\begin{aligned}N &= \frac{{\rho_u}^{n_{th1}+2}\left(\left(n_{th1}+1\right)\rho_u - n_{th1} - 2\right)}{\left(\rho_u-1\right)^2} + \frac{\rho_u}{\left(\rho_u-1\right)^2} \\ &\quad +\frac{{\rho_u}^{n_{th1}+1}{\rho_r}^{n_{th2}-n_{th1}+1}\left(\left(n_{th2}+1\right)\rho_r - n_{th2} - 2\right)}{\left(\rho_r-1\right)^2} \\ &\quad -\frac{{\rho_u}^{n_{th1}+1}\rho_r\left(\left(n_{th1}+1\right)\rho_r - n_{th1} - 2\right)}{\left(\rho_r-1\right)^2} \\ &\quad +\frac{{\rho_u}^{n_{th1}+1}{\rho_r}^{n_{th2}-n_{th1}}{\rho_m}^{n-n_{th2}}\left(n\rho_m - n - 1\right)}{\left(\rho_m-1\right)^2} \\ &\quad -\frac{{\rho_u}^{n_{th1}+1}{\rho_r}^{n_{th2}-n_{th1}}\rho_m\left(\left(n_{th2}+1\right)\rho_m - n_{th2} - 2\right)}{\left(\rho_m-1\right)^2}\end{aligned}$$

As before, we assume that enough buffer space is allocated above the n_{th2} threshold, so that committed traffic loss probability is acceptable and in agreement with the SLA (and this is the case in the following figures); But the expressions given above do without this assumption and are valid for the general case where the buffer is finite.

Three figures are included to demonstrate the behavior of the system in this case. Figure 14 shows the total buffer occupancy distribution function. Figure 15 shows the new trade off between the acceptance ratios of the two excess traffic classes, controlled by the position of the n_{th2} threshold. Note that acceptance ratio for the high priority excess traffic reaches 100% since the aggregate load of the committed traffic (both high and low priority) and the high priority excess traffic is lower than unity. Finally figure 16 show the effect of n_{th2} on the low priority expected delay.

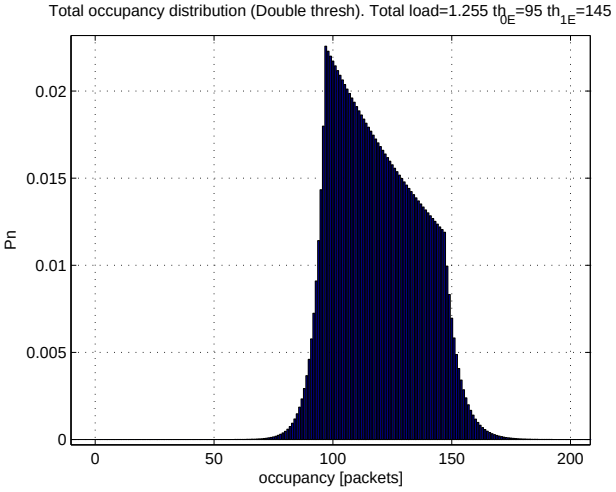

Fig. 14. Total Occupancy distribution function. Traffic loads are $\lambda 1 = \lambda 3 = 0.33$ for committed packet streams, high and low priority excess traffic loads are $\lambda 2 = 0.12$ and $\lambda 4 = 0.21$ respectively.

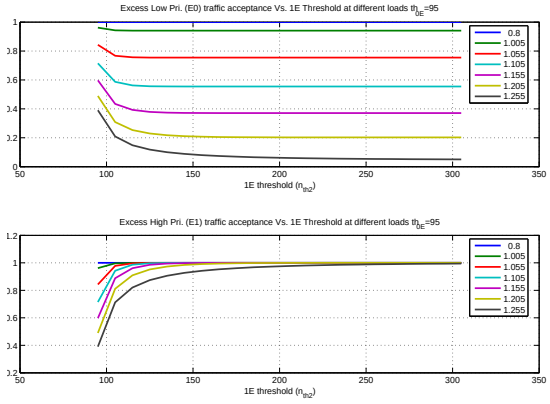

Fig. 15. Excess Traffic Acceptance vs. High priority excess traffic threshold. Traffic loads are $\lambda 1 = \lambda 3 = 0.33$ for committed packet streams, high and low priority excess traffic loads are $\lambda 2 = 0.12$ and $\lambda 4 = 0.21$ respectively.

VIII. Concluding Remarks

References

[AMRR00] W. Aiello, Y. Mansour, S. Rajagopolan, and A. Rosen. Competitive queue policies for diffrentiated services. In *IEEE INFOCOM*, Tel Aviv, Israel, March 2000.

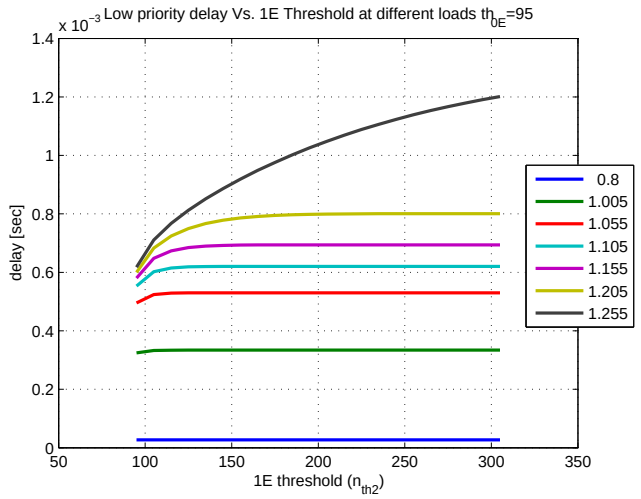

Fig. 16. Low Priority traffic delay vs. High priority excess traffic threshold. Traffic loads are $\lambda 1 = \lambda 3 = 0.33$ for committed packet streams, high and low priority excess traffic loads are $\lambda 2 = 0.12$ and $\lambda 4 = 0.21$ respectively.

[BBC+98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated services. Internet RFC 2475, November 1998.

[CGGK95] Israel Cidon, Leonidas Georgiadis, Roch Guerin, and Asad Khamisy. Optimal buffer sharing. *IEEE Journal on Selected Areas in Communications*, 13(7):1229–1240, September 1995.

[CGK94] Israel Cidon, Roch Guerin, and Asad Khamisy. On protective buffer policies. *IEEE/ACM Transactions on Networking*, 2(3):240–246, June 1994. special issue on "Advances in the Fundamentals of Networking".

[CH98] Abhijit K. Choudhury and Ellen L. Hahne. Dynamic queue length thresholds for shared-memory packet switches. *IEEE/ACM Transactions on Networking*, 6(2):130–140, 1998.

[For04] Metro Ethernet Forum. Ethernet services attributes phase 1. Metro Ethernet Technical Specifications, November 2004.

[FT89] Gregory L. Frazier and Yuval Tamir. The design and implementation of a multi-queue buffer for vlsi communication switches. In *International Conference on Computer Design*, pages 466–471, Cambridge, MA, USA, October 1989.

[FWD+02] F. Le Faucheur, L. Wu, B. Davie, S. Davari, P. Vaananen, R. Krishnan, P. Cheval, and J. Heinanen. Multi-protocol label switching (mpls)support of differentiated services. Internet RFC 3270, May 2002.

[Gro02] D. Grossman. New terminology and clarifications for diffserv, April 2002. Internet RFC 3260.

[IKM01] Sundar Iyer, Ramana Rao Kompella, and Nick McKeown. Analysis of a memory architecture for fast packet buffers. In *IEEE Workshop on High Performance Switching and Routing*, pages 466–471, Dallas, TX, USA, May 2001.

[KK80] F. Kamoun and L. Kleinrock. Analysis of shared finite storage in a computer network node environment under general traffic conditions. *IEEE trans. on commun.*, COM-28:992–1003, 1980.

[Kle76] L. Kleinrock. *Queuing Systems, Vol. 2: Computer Applications.* John Wiley and Sons, Inc., 1976.

[LPS02] Zvi Lotker and Boaz Patt-Shamir. Nearly optimal FIFO buffer management for DiffServ. In *PODC'02*, Monterey, CA, USA, July 2002.

[NBBB98] K. Nichols, S. Blake, F. Baker, and D. Black. Definition of the differentiated services field (DS field) in the IPv4 and IPv6 headers, December 1998. Internet RFC 2474.

[NC01] K. Nichols and B. Carpenter. Definition of differentiated services per domain behaviors and rules for their specification, April 2001. Internet RFC 3086.

[San04] Ralph Santitoro. Bandwidth profiles for ethernet servicesf. Metro Ethernet Forum White Paper, January 2004.