Analysis of Shared Memory Priority Queues with Two Discard Levels

Shlomi Bergida and Yuval Shavitt
School of Electrical Engineering
Tel Aviv University

***Abstract*— Two rate SLAs become increasingly popular in today's Internet, allowing a customer to save money by paying one price for committed 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. In access networks, where these SLAs are prevalent, shared memory switches is a common architecture. Thus, dimensioning and management of shared memory queues for multiple priorities each with two levels of guarantees becomes an interesting challenge.**

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 assuring high service level agreement compliance for conforming traffic, and performance maximization for non-conforming 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 [15], [4], [16], [10]. They were later also incorporated into the MPLS world in the form of MPLS DiffServ-TE technology [8], and recently introduced into the Metro Ethernet world, with the standardization efforts of the Metro Ethernet Forum [1], [17].

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 during congestion 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 during congestion periods typically includes a packet drop policy to manage buffer space.

Queue management has been studied extensively [12], [5] 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 [9], [11].

There is a long line of work that examines threshold policies for two (or more) types of packets that share a single FIFO buffer [7], [6], [2], [14], 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 switch 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 [7]).

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 Fig. 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.

To allow simple and efficient analysis and calculation, we present analysis that uses a simpler model, where 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. Thus, in part of our analysis 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 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.

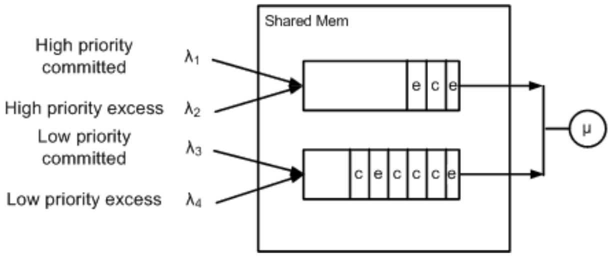

Fig. 1. System model

II. Exact analysis

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.

An exact analysis of the above system can be done by using a continuous-time two-dimensional Markov chain with $(n+1)(n+2)/2$ states, where 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 (see Fig. 2). The system can be solved in $O(n^3)$ computations yielding the delay, buffer occupancy, and throughput of each traffic class.

III. Approximated Analysis

The complexity of the exact numerical solution may be too high to allow solving the system for buffer

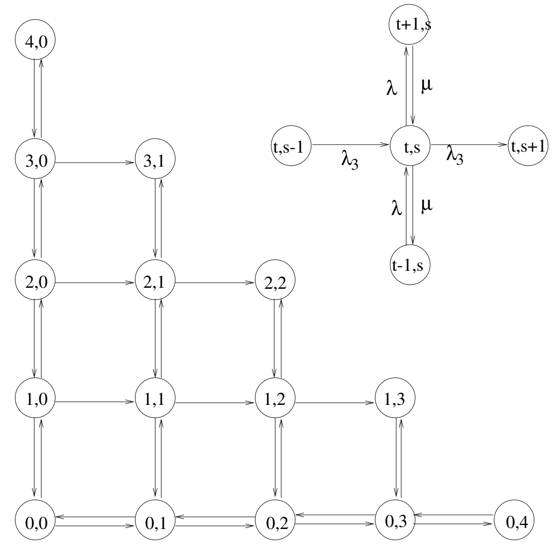

Fig. 2. A Markov chain for the two queue system.

sizes exceeding a few tens of packets. For the case where system behavior is controlled by total occupancy thresholds we suggest instead to first model the system by a single parameter, its total occupancy, as explained below. Using this value we then derive the other system parameters with a finer analysis.

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.

In this analysis the system can be modeled by a continuous-time birth-death Markov chain with $n+1$ states. The state transition probabilities are given by

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

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

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

$$q_{u,u+1}\pi_u = q_{u,u-1}\pi_{u+1} \qquad 0 \le u \le n-1 \qquad (2)$$

$$\sum_{u=0}^{n} \pi_u = 1 \qquad (3)$$

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

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

Solving this equation explicitly yields

$$\pi_u = \begin{cases} \pi_0(\rho_u)^u & \text{if } u \le n_{th}+1 \\ \pi_{(n_{th}+1)}(\rho_r)^{(u-n_{th}-1)} & \text{if } n_{th}+1 < u \le n \end{cases} \qquad (5)$$

and applying the probability conservation (eq. 3) yields

$$\pi_0 = \frac{(1-\rho_u)(1-\rho_r)}{(1-\rho_r)(1-\rho_u^{n_{th}+2}) + \rho_u^{n_{th}+1}\rho_r(1-\rho_u)(1-\rho_r^{n-n_{th}-1})} \qquad (6)$$

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:

$$\begin{aligned} \eta_1 &= \pi_n \\ \eta_2 &= \sum_{u=n_{th}+1}^{n} \pi_u \qquad (7) \\ \eta_3 &= \pi_n \end{aligned}$$

explicitly:

$$\begin{aligned} \eta_1 &= \pi_0 \rho_u^{n_{th}+1} \rho_r^{n-n_{th}-1} \\ \eta_2 &= \pi_0 \rho_u^{n_{th}+1} \frac{1-\rho_r^{n-n_{th}}}{1-\rho_r} \qquad (8) \\ \eta_3 &= \eta_1 \end{aligned}$$

We are interested in the regime where $\rho_r = (\lambda_1 + \lambda_3)/\mu < 1$ 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 3 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 4 shows the acceptance ratio 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 graphs include simulation results for comparison.

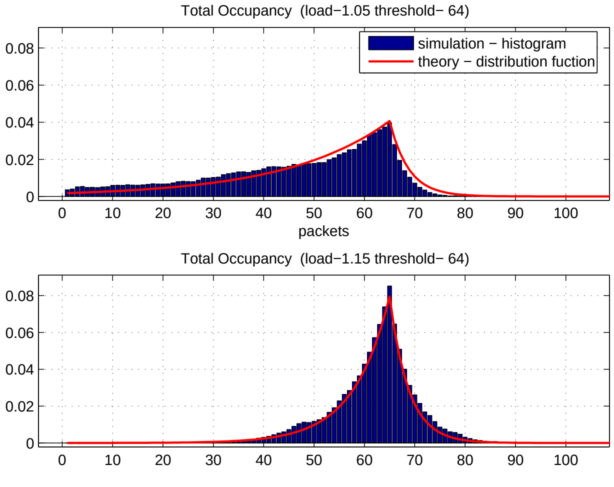

Fig. 3. Total system occupancy distribution function at two load points

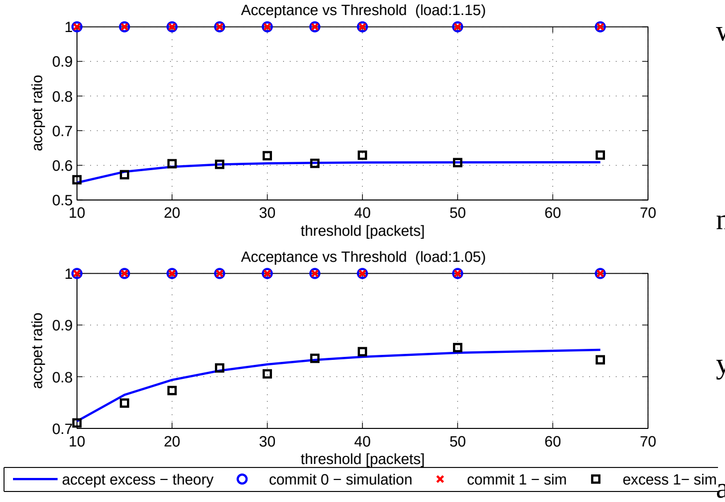

Fig. 4. Acceptance Vs. Threshold at two load points. The buffer size is large enough so committed traffic loss probability is negligable

B. *Delay analysis*

For our approximated analysis of the delay in the threshold governed priority queue we will use an approach based on the multi-priority with infinite capacity analysis of Kleinrock [13].

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 above.

The waiting time equation for priority 1 for our case is

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

substituting $\hat{\lambda}_1 W_1$ for N_1 by Little's theorem and rearranging yields:

$$W_1 \quad = \quad \frac{\hat{W_0}}{1 - \bar{x_1}\hat{\lambda_1}} \tag{9}$$

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.[1] The chance of the server being free is $1 - \pi_0$ (see equation 6). Therefore $\hat{W_0}$ can be written as:

$$\hat{W_0} \quad = \quad \frac{1}{\mu}(1 - \pi_0) \tag{10}$$

Since we consider infinite total capacity, the total system occupancy distribution function (equation 5) is

$$\begin{aligned} \pi_u = & \quad \pi_0(\rho_u)^u & \text{if } u \leq n_{th} + 1 \\ \pi_u = & \quad \pi_{n_{th}}(\rho_r)^{(u - n_{th})} & \text{if } n_{th} + 1 < u \end{aligned} \tag{11}$$

where

$$\pi_0 = \frac{(1-\rho_u)(1-\rho_r)}{(1-\rho_r)(1-\rho_u^{n_{th}+2}) + \rho_u^{n_{th}+1}\rho_r(1-\rho_u)} \tag{12}$$

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

$$\begin{aligned} \hat{\lambda_1} \quad = & \quad (\lambda_1 + \lambda_2) \cdot P(occupancy \leq n_{th}) \\ & \quad + \lambda_1 \cdot P(n_{th} < occupancy) \end{aligned} \tag{13}$$

yielding :

$$\hat{\lambda_1} \quad = \quad \pi_0 [\frac{1 - \rho_u^{n_{th}+1}}{1 - \rho_u}(\lambda_1 + \lambda_2) + \frac{\rho_u^{n_{th}+1}}{1 - \rho_r}\lambda_1] \tag{14}$$

and thus, we have:

$$W_1 \quad = \quad \frac{\hat{W_0}}{1 - \frac{1}{\mu}\hat{\lambda_1}} \tag{15}$$

Using equations 10, 12, 14, and 15 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:

$$N_0 \quad = \quad \hat{\lambda_0} W_0 = N - N_1 \tag{16}$$

$N_1 = W_1 \hat{\lambda_1}$, and N can be calculated using the results of the total system occupancy distribution (eq. 11), yielding

$$\begin{aligned} N \quad = & \quad \frac{\pi_0}{(1-\rho_u)^2}[\rho_u^{n_{th}+2}((n_{th}+1)\rho_u - (n_{th}+2)) + \rho_u] - \\ & \quad \frac{\pi_0}{(1-\rho_r)^2}[\rho_r \rho_u^{n_{th}+1}((n_{th}+1)\rho_r - (n_{th}+2))] \end{aligned} \tag{17}$$

[1] Clearly, the model allows for traffic in each priority to have different stochastic characteristics.

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

$$W_0 = \frac{N - W_1\hat{\lambda_1}}{\lambda_3} \tag{18}$$

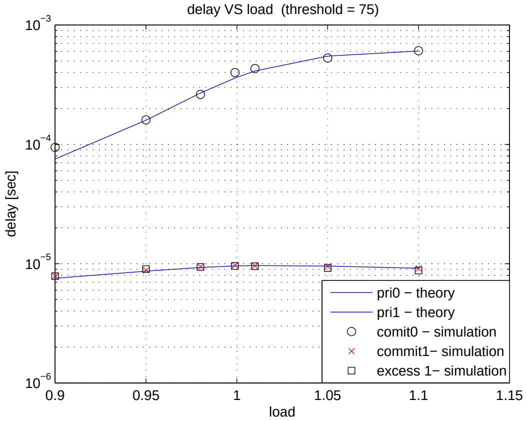

Fig. 5. Delay vs. Load

In Fig. 5 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.

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, the committed aggregate input rate, $\lambda_1 + \lambda_3$, is lower than the service rate, and buffer space above the threshold, $n - n_t h$, 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 7) are negligible. Graphs and analysis were removed due to space limitations; for more details see [3].

IV. Concluding Remarks

This paper is a first step in understanding the management of multi-priority queue where traffic for each priority queue is comprised of two discard levels. We used here queueing theory to derive expressions for the expected delay and throughput of a simple threshold policy. We believe that continuing this direction, as well as using other tools such as competitive analysis, will help us better understand how to engineer such systems.

References

[1] Ethernet services attributes phase 1. Metro Ethernet Technical Specifications, Nov. 2004.

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

[3] S. Bergida and Y. Shavitt. Analysis of shared memory priority queues with two discard levels. Technical Report EE60, Tel Aviv University, School of Electrical Engineering, Sept. 2006.

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

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

[6] I. Cidon, L. Georgiadis, R. Guerin, and A. Khamisy. Optimal buffer sharing. *IEEE Journal on Selected Areas in Communications*, 13(7):1229–1240, Sept. 1995.

[7] I. Cidon, R. Guerin, and A. 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”.

[8] F. L. 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.

[9] G. L. Frazier and Y. 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, Oct. 1989.

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

[11] S. Iyer, R. R. Kompella, and N. 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.

[12] 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.

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

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

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

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

[17] R. Santitoro. Bandwidth profiles for Ethernet services. Metro Ethernet Forum White Paper, Jan. 2004.