

On priority queues with priority jumps

Tom Maertens, Joris Walraevens and Herwig Bruneel

Ghent University – UGent
Department of Telecommunications and Information Processing (IR07)
SMACS Research Group
Sint-Pietersnieuwstraat 41, B-9000 Gent, Belgium
Phone: +32-9-2648901
Fax: +32-9-2644295
E-mail: tmaerten@telin.ugent.be

Abstract

In this paper, we investigate a simplified head-of-the-line with priority jumps (HOL-PJ) scheduling discipline. Therefore, we consider a discrete-time single-server queueing system with two priority queues of infinite capacity and with a newly introduced HOL-PJ priority scheme. We derive expressions for the probability generating function of the system contents and the packet delay. Some performance measures (such as mean and variance) of these quantities are derived and are used to illustrate the impact and significance of the HOL-PJ priority scheduling discipline in an output queueing switch. We compare this dynamic priority scheduling discipline with a FIFO scheduling and a static priority scheduling (HOL) and we investigate the influence of the different parameters of the simplified HOL-PJ scheduling discipline.

Keywords

discrete-time, queueing theory, dynamic priority scheduling, performance analysis

1 Introduction

One of the main keys to a successful network is the ability to support different applications. Different applications generate different types of traffic, which need different QoS (Quality of Service) standards. For real-time applications, it is important that mean delay and delay variance are

not too large, while for non real-time applications, the packet loss ratio is the restrictive quantity. To support multiple classes of traffic in multimedia networks, priority mechanisms can be used. Multiple priority levels are provided, and different priority levels are given to different classes of traffic. There are two ways to use priorities; the first is to use a priority mechanism as a scheduling method (or a queueing discipline). Different delay requirements can be satisfied by scheduling delay-sensitive traffic first. The second way is to use a priority scheme to control congestion. In this case, when network congestion occurs, different packet loss requirements can be satisfied by selectively discarding low-priority packets. In this paper, we will only focus on a priority scheme as a scheduling method and we will thus not discuss local congestion control schemes.

The simplest priority scheme is a static priority scheme. In this discipline, priority is always given to the delay-sensitive class, and thus packets of this class are always scheduled for service before delay-insensitive packets. This scheme provides relatively low delays for the delay-sensitive class (see e.g. [1, 5, 9, 10, 13, 15]). But if a large portion of the network traffic consists of high-priority traffic, the performance for the low-priority traffic can be severely degraded. Specifically, head-of-line (HOL) priority scheduling can cause excessive delays for the low-priority class, especially if the network is highly loaded. This problem is also known as the starvation problem. In order to find a solution for this problem, several dynamic priority schemes have been proposed in the literature. A class of dynamic priority schemes are the queue-length-threshold scheduling disciplines (QLT). In these disciplines, the priority queues are served depending on their queue length (analyses, see [4, 6, 7, 8, 11, 12]). Another class of dynamic priority schemes are the head-of-line with priority jumps (HOL-PJ) disciplines. This discipline was introduced in [14]. In a queueing system with this type of discipline, high-priority packets arrive in a high-priority queue, while low-priority packets arrive in a low-priority queue (and the packets in the high-priority queue thus have service priority over packets in the low-priority queue). However, packets (at the head) of the low-priority queue can jump to (the end of) the high-priority queue. Many criteria can be used to decide when low-priority packets jump, e.g., a maximum queueing delay

in the low-priority queue (see [14]), a queue-length threshold of the low-priority queue, a random jumping probability per time unit,...

In [14] e.g., a limit is imposed on the maximum queueing delay for the packets in the low-priority queue. The value of this limit is set to the *difference* between the delay requirement for the low-priority class and that for the high-priority class. Immediately after a packet's delay at the low-priority queue equals (or exceeds) the limit, the packet jumps to the tail of the high-priority queue. The jumped packet is treated in the high-priority queue as if it was a newly arriving packet to that queue. This jumping mechanism guarantees that at any time, the packets are queued according to the largeness of the queueing delays in *excess* of their respective delay requirements. Possible disadvantages of this scheme are the processing overhead required for monitoring packets for time-out and the additional implementation hardware, i.e., a local clock at the low-priority queue, a list of the arrival times of the low-priority packets, and circuitry for the timing mechanism to make the packet(s) at the head of the queue jump.

In this paper, we alter this HOL-PJ scheme, mainly in order to evade these disadvantages. We introduce a parameter β , which yields the probability that the total contents of the low-priority queue jumps to the end of the high-priority queue. The value of β can be chosen in such a way that the low-priority traffic stays within its delay requirements, i.e., the more stringent the delay requirement, the larger the value of β . Also, when $\beta = 0$ in our model, this is the well-known static priority queue (which has been fully analyzed in [15]).

We have chosen to analyze this type of priority jumping for a number of reasons. Firstly, this scheme can be implemented relatively easy. Indeed, it is not necessary to keep track of the queueing delays of the low-priority packets, which makes a major difference with regard to the implementation (i.e., no clock, no list or no timing mechanism). Secondly, the analysis is kept analytically tractable. We use an analysis based on generating functions for assessing the performance of queues with the introduced HOL-PJ scheduling discipline. We obtain the joint generating function of the contents in the high- and low-priority queues and the generating functions of the delay of high- and low-priority packets. From the obtained generating functions,

we can then easily calculate expressions for some interesting performance measures, such as mean value and variance of the queue contents and packet delay. These expressions require little computational effort and are well-suited for evaluating the impact of the various system parameters on the overall performance. We will also show that our results can be applied to the case of an output-queueing switch (see section 6) with a HOL-PJ priority scheduling. Thirdly, we will show that the delay of high- and low-priority packets in this scheme and the scheme of [14] are comparable. By means of simulation results, we will show that the performance of the output-queueing switch for the simplified scheme hardly differs from the performance for the original HOL-PJ scheme of [14], with respect to the (mean) packet delay. This makes the newly introduced scheme quite attractive.

The contribution of this paper concerns the model that is considered, as well as the solution technique that is used. First, as far as the model is concerned, the main difference with most of the articles mentioned in this introduction is that, for the case of a multi-class output-queueing switch, the arrival processes of the different types of packets are not mutually independent, which complicates the analysis. Secondly, as discussed before, we 'summarize' the jumping process by a single parameter β . Finally, we want to show that a generating-function solution method is suitable for analyzing this type of buffers with a HOL-PJ priority scheduling.

The outline of the paper is as follows. In the following section, we describe the mathematical model. In sections 3 and 4, we analyze the steady-state system contents and packet delay of both classes. In section 5, we calculate the moments of the system contents and packet delay. We apply the obtained results to an output-queueing switch with Bernoulli arrivals, and discuss the impact of the HOL-PJ priority scheduling discipline in section 6. Some conclusions are formulated in section 7.

2 Mathematical model

In the next sections, we investigate a discrete-time queueing system with one server and two queues whereby the capacity of the queues is infinite. Time is assumed to be slotted and the transmission time of a packet is one slot. There are two types of traffic arriving in the system, namely packets of class 1 and packets of class 2, which arrive in the first and the second queue respectively. The number of arrivals of class j during slot k is denoted by $a_{j,k}$ ($j = 1, 2$) and the $a_{j,k}$'s are independent and identically distributed (i.i.d.) from slot-to-slot. However, in one slot, the number of arrivals of one class can be correlated with the number of arrivals of the other class. This dependence is characterized by their joint probability mass function (pmf) $a(m, n)$:

$$a(m, n) \triangleq \text{Prob}[a_{1,k} = m, a_{2,k} = n],$$

and by the corresponding joint probability generating function (pgf) $A(z_1, z_2)$:

$$A(z_1, z_2) \triangleq \text{E} [z_1^{a_{1,k}} z_2^{a_{2,k}}].$$

The total number of arriving packets during slot k is denoted by $a_{T,k} \triangleq a_{1,k} + a_{2,k}$ and its pgf is defined as $A_T(z) \triangleq \text{E} [z^{a_{T,k}}] = A(z, z)$. Further, we define the marginal pgf's of the number of arrivals from class 1 and class 2 during a slot as $A_1(z) \triangleq \text{E} [z^{a_{1,k}}] = A(1, z)$ and $A_2(z) \triangleq \text{E} [z^{a_{2,k}}] = A(z, 1)$ respectively. From these pgf's, we can calculate the arrival rate of class j : $\lambda_j \triangleq \text{E} [a_{j,k}] = A'_j(1)$. The total arrival rate is the sum of the arrival rates of both classes: $\lambda_T = A'_T(1) = \lambda_1 + \lambda_2$.

The system has one server that provides the transmission of packets, at a rate of one packet per slot. Newly arriving packets can enter service at the beginning of the slot following their arrival slot at the earliest. Cells in queue 1 have a higher priority than those in queue 2. Hence, when there are packets in the first queue, they have service priority and only when the first queue is empty, packets in the second queue can be served. Within a queue, the service discipline is

FIFO.

In each slot, the contents of queue 2 jumps to queue 1 with probability β . The jumps occur at the end of the slot and the jumping packets are scheduled after the contents of the first queue at the end of the slot.

3 System contents

In this section, we derive the steady-state joint pgf of the system contents of both queues. We assume that the packet in service (if any) is part of the queue that is serviced in that slot. We denote the system contents of queue j at the beginning of slot k by $u_{j,k}$ and the total system contents at the beginning of slot k by $u_{T,k}$. The joint pgf of $u_{1,k}$ and $u_{2,k}$ is denoted by $U_k(z_1, z_2)$:

$$U_k(z_1, z_2) \triangleq \mathbb{E} [z_1^{u_{1,k}} z_2^{u_{2,k}}].$$

The system contents of both queues evolves in time according to the following system equations:

- if $u_{1,k} = 0$:

$$u_{1,k+1} = \begin{cases} a_{1,k} + [u_{2,k} - 1]^+ + a_{2,k} & \text{with probability } \beta \\ a_{1,k} & \text{with probability } 1 - \beta \end{cases}, \quad (1)$$

$$u_{2,k+1} = \begin{cases} 0 & \text{with probability } \beta \\ [u_{2,k} - 1]^+ + a_{2,k} & \text{with probability } 1 - \beta \end{cases}, \quad (2)$$

- if $u_{1,k} > 0$:

$$u_{1,k+1} = \begin{cases} u_{1,k} - 1 + a_{1,k} + u_{2,k} + a_{2,k} & \text{with probability } \beta \\ u_{1,k} - 1 + a_{1,k} & \text{with probability } 1 - \beta \end{cases}, \quad (3)$$

$$u_{2,k+1} = \begin{cases} 0 & \text{with probability } \beta \\ u_{2,k} + a_{2,k} & \text{with probability } 1 - \beta \end{cases}, \quad (4)$$

where $[\dots]^+$ denotes the maximum of the argument and zero. System equations (1) and (2) can be explained as follows: if queue 1 is empty at the beginning of slot k , a packet of queue 2 (if any) is served during slot k . The remaining packets of queue 2, plus the arriving packets of class 2, jump with a probability β to queue 1. On the other hand, if there is at least one packet present in queue 1 at the beginning of slot k , a packet of class 1 is served (equations (3) and (4)). Also in this situation, the contents of queue 2 and the arriving packets of class 2, jump with a probability β to queue 1 at the end of slot k .

Calculation of the joint pgf of the system contents of both queues at the beginning of slot $k + 1$, yields

$$\begin{aligned} U_{k+1}(z_1, z_2) &\triangleq \mathbb{E} [z_1^{u_{1,k+1}} z_2^{u_{2,k+1}}] \\ &= \mathbb{E} [z_1^{u_{1,k+1}} z_2^{u_{2,k+1}} \{u_{1,k} = 0\}] \\ &\quad + \mathbb{E} [z_1^{u_{1,k+1}} z_2^{u_{2,k+1}} \{u_{1,k} > 0\}] \\ &= \beta \mathbb{E} [z_1^{u_{1,k+1}} z_2^{u_{2,k+1}} \{u_{1,k} = 0\}] \\ &\quad + (1 - \beta) \mathbb{E} [z_1^{u_{1,k+1}} z_2^{u_{2,k+1}} \{u_{1,k} = 0\}] \\ &\quad + \beta \mathbb{E} [z_1^{u_{1,k+1}} z_2^{u_{2,k+1}} \{u_{1,k} > 0\}] \\ &\quad + (1 - \beta) \mathbb{E} [z_1^{u_{1,k+1}} z_2^{u_{2,k+1}} \{u_{1,k} > 0\}], \end{aligned} \quad (5)$$

with $\mathbb{E} [X \{Y\}] \triangleq \mathbb{E} [X|Y] \text{Prob} [Y]$. Using the system equations, we form the following relation between $U_{k+1}(z_1, z_2)$ and $U_k(z_1, z_2)$:

$$\begin{aligned} U_{k+1}(z_1, z_2) &= \beta A(z_1, z_1) \left\{ \frac{z_2(z_1 - 1)U_k(0, 0) + z_2 U_k(z_1, z_1)}{z_1 z_2} \right\} \\ &\quad + (1 - \beta) A(z_1, z_2) \left\{ \frac{z_1(z_2 - 1)U_k(0, 0) + (z_1 - z_2)U_k(0, z_2) + z_2 U_k(z_1, z_2)}{z_1 z_2} \right\}. \end{aligned} \quad (6)$$

Since we are interested in the steady-state distribution of the system contents, we define $U(z_1, z_2)$ as:

$$U(z_1, z_2) \triangleq \lim_{k \rightarrow \infty} U_k(z_1, z_2).$$

Applying this limit in equation (6) and solving the resulting equation for $U(z_1, z_2)$, we find

$$\begin{aligned} U(z_1, z_2) &= \beta A(z_1, z_1) \left\{ \frac{z_2(z_1 - 1)U(0, 0) + z_2 U(z_1, z_1)}{z_2(z_1 - (1 - \beta)A(z_1, z_2))} \right\} \\ &+ (1 - \beta)A(z_1, z_2) \left\{ \frac{z_1(z_2 - 1)U(0, 0) + (z_1 - z_2)U(0, z_2)}{z_2(z_1 - (1 - \beta)A(z_1, z_2))} \right\}. \end{aligned} \quad (7)$$

In the right hand side of equation (7), there are three quantities yet to be determined, namely the functions $U(0, z_2)$ and $U(z_1, z_1)$ and the constant $U(0, 0)$. First, we compute the function $U(z_1, z_1)$. Substitution of z_2 by z_1 in expression (7) yields:

$$U(z_1, z_1) = U(0, 0) \frac{A_T(z_1)(z_1 - 1)}{z_1 - A_T(z_1)}. \quad (8)$$

Notice that $U(z, z) = U_T(z) \triangleq \mathbb{E}[z^{u_T, k}]$, the pgf of the total system contents. This expression is identical to the generating function of the system contents of a queue with a FCFS-discipline and with one class (with arrivals determined by $A_T(z)$). This is expected, because for the total system contents, it does not matter in which order the packets are being served. Next, we can determine the constant $U(0, 0)$ from equation (8) by substituting z_1 by 1, by applying the normalization condition $U(1, 1) = 1$ and by using l'Hopital's rule. The result is the probability of having an empty system: $U(0, 0) = 1 - \lambda_T$. Notice that the stability condition equals $\lambda_T < 1$, since the queueing system is work-conserving and λ_T equals the (arrival) load of the system.

Finally, we derive an expression for the function $U(0, z_2)$. By applying Rouché's theorem, it can be proven that for a given value of z_2 in the unit circle ($|z_2| \leq 1$), the equation $z_1 = (1 - \beta)A(z_1, z_2)$ has one solution in the unit circle for z_1 , which will be denoted by $Y(\beta, z_2)$ in the remainder, and which is implicitly defined by $Y(\beta, z) \triangleq (1 - \beta)A(Y(\beta, z), z)$. Since $Y(\beta, z_2)$ is a zero of the denominator of the right hand side of (7) and since a generating function remains

finite in the unit circle, $Y(\beta, z_2)$ must also be a zero of the numerator. We thus find:

$$\begin{aligned}
U(0, z_2) &= \beta A_T(Y(\beta, z_2)) \left\{ \frac{U(0, 0)z_2(Y(\beta, z_2) - 1) + z_2 U_T(Y(\beta, z_2))}{Y(\beta, z_2)(z_2 - Y(\beta, z_2))} \right\} \\
&+ \frac{Y(\beta, z_2)(z_2 - 1)U(0, 0)}{z_2 - Y(\beta, z_2)}. \tag{9}
\end{aligned}$$

Using expressions (8) and (9) and the expression for $U(0, 0)$ in equation (7) finally yields $U(z_1, z_2)$:

$$\begin{aligned}
U(z_1, z_2) &= \frac{(1 - \lambda_T)\beta A_T(z_1)z_1(z_1 - 1)}{(z_1 - A_T(z_1))(z_1 - (1 - \beta)A(z_1, z_2))} \\
&+ \frac{(1 - \lambda_T)(1 - \beta)A(z_1, z_2)}{(z_1 - (1 - \beta)A(z_1, z_2))(z_2 - Y(\beta, z_2))} \\
&\times \left\{ (z_2 - 1)(z_1 - Y(\beta, z_2)) + \frac{(z_1 - z_2)\beta A_T(Y(\beta, z_2))(Y(\beta, z_2) - 1)}{Y(\beta, z_2) - A_T(Y(\beta, z_2))} \right\}. \tag{10}
\end{aligned}$$

From this joint pgf, we can calculate the marginal pgfs $U_j(z)$ ($j = 1, 2$) of the system contents of class j :

$$\begin{aligned}
U_1(z) &\triangleq \lim_{k \rightarrow \infty} \mathbb{E}[z^{u_1, k}] = U(z, 1) \\
&= \frac{(1 - \lambda_T)\beta A_T(z)z(z - 1)}{(z - A_T(z))(z - (1 - \beta)A_1(z))} \\
&- \frac{(1 - \lambda_T)(1 - \beta)\beta A_T(Y(\beta, 1))A_1(z)(z - 1)}{(z - (1 - \beta)A_1(z))(Y(\beta, 1) - A_T(Y(\beta, 1)))}, \tag{11}
\end{aligned}$$

$$\begin{aligned}
U_2(z) &\triangleq \lim_{k \rightarrow \infty} \mathbb{E}[z^{u_2, k}] = U(1, z) \\
&= \frac{(1 - \lambda_T)(1 - \beta)A_2(z)(z - 1)(1 - Y(\beta, z))}{(1 - (1 - \beta)A_2(z))(z - Y(\beta, z))} \frac{Y(\beta, z) - (1 - \beta)A_T(Y(\beta, z))}{Y(\beta, z) - A_T(Y(\beta, z))} \\
&+ \frac{\beta}{1 - (1 - \beta)A_2(z)}. \tag{12}
\end{aligned}$$

It is mentioned in the introduction that when $\beta = 0$, the investigated scheduling discipline equals a static priority scheme. Indeed, by substituting $\beta = 0$ in expressions (10), (11) and (12), and by using l'Hopital's rule and the definition of $Y(\beta, z)$, we obtain the same expressions as in

[15], for $U(z_1, z_2)$, $U_1(z)$ and $U_2(z)$. $U(z_1, z_2)$ for instance equals

$$U(z_1, z_2) = (1 - \lambda_T) \frac{A(z_1, z_2)(z_2 - 1)(z_1 - Y(0, z_2))}{(z_2 - Y(0, z_2))(z_1 - A(z_1, z_2))},$$

with $Y(0, z) = A(Y(0, z), z)$, when $\beta = 0$.

4 Packet delay

The packet delay is defined as the total amount of time that a packet spends in the system, i.e., the number of slots between the end of the packet's arrival slot and the end of its departure slot. In this section, we will derive expressions for the pgf's of the packet delay of both classes.

Since a jump of the contents of queue 2 to queue 1 takes place at the end of the slot, the newly arriving packets of class 1 are queued in front of packets that jump in the same slot. As a consequence, the packet delay of a tagged class 1 packet only depends on the system contents of queue 1 at the beginning of its arrival slot. This also means that the packet delay of a tagged class 1 packet is the same as in a corresponding single-class system with only packets of class 1 arriving. The amount of time a tagged class 1 packet spends in the system equals

$$d_1 = [u_{1,k} - 1]^+ + f_{1,k} + 1, \tag{13}$$

where slot k is assumed to be the arrival slot of the tagged packet, $u_{1,k}$ is the system contents of queue 1 at the beginning of this slot and $f_{1,k}$ is defined as the total number of class 1 packets that arrive during slot k , and which have to be served before the tagged packet. Indeed, the tagged class 1 packet has to wait in queue 1 until all packets that were already in this queue at the moment of its arrival, are effectively served. The number of these packets is obviously determined by all packets that were already present in queue 1 at the beginning of its arrival slot (potentially including class 2 packets which jumped to queue 1 before the tagged packet arrived) and all class 1 packets that arrived before the tagged packet in its arrival slot. The delay then

equals this waiting time augmented with the service time of a packet, which equals 1. This leads to expression (13). Introducing pgf's yields

$$\begin{aligned} D_1(z) &\triangleq \mathbb{E}[z^{d_1}] \\ &= F_1(z) [U_1(z) + (z-1)U_1(0)]. \end{aligned} \quad (14)$$

Notice that we have used the fact that $u_{1,k}$ and $f_{1,k}$ are uncorrelated (because the numbers of arrivals are i.i.d. from slot-to-slot) to obtain (14). The pgf $F_1(z) \triangleq \mathbb{E}[z^{f_{1,k}}]$ can be calculated by taking into account that an arbitrary packet is more likely to arrive in a larger bulk (e.g. [2]), yielding

$$F_1(z) = \frac{A_1(z) - 1}{\lambda_1(z-1)}. \quad (15)$$

Using expressions (11) and (15) in (14), we finally find

$$\begin{aligned} D_1(z) &= \frac{(1-\lambda_T)\beta z A_T(z)(A_1(z)-1)}{\lambda_1(z-A_T(z))(z-(1-\beta)A_1(z))} - \frac{(1-\lambda_T)\beta(1-\beta)A_T(Y(\beta,1))A_1(z)(A_1(z)-1)}{\lambda_1(z-(1-\beta)A_1(z))(Y(\beta,1)-A_T(Y(\beta,1)))} \\ &\quad - \frac{(1-\lambda_T)\beta A_T(Y(\beta,1))(A_1(z)-1)}{\lambda_1(Y(\beta,1)-A_T(Y(\beta,1)))}. \end{aligned} \quad (16)$$

The analysis of the packet delay of a class 2 packet is more complicated. Consider a logically equivalent queueing system where all high-priority packets of queue 1 are stored in front of the packets of queue 2, and let us tag an arbitrary class 2 packet that arrives in the system. The amount of time it spends in the system equals

$$d_2 = \sum_{j=1}^{\lceil u_{T,k}-1 \rceil + \hat{f}_{2,k}} v_j^0 + 1, \quad (17)$$

where slot k is assumed to be the arrival slot of the tagged packet, $u_{T,k}$ is the total system contents at the beginning of slot k and $\hat{f}_{2,k}$ is defined as the total number of packets that arrive during the arrival slot of the tagged packet, but which have to be served before it. However, the

most important quantity here is v_j^0 . This quantity represents the number of slots it takes for the tagged packet to move one position ahead in the queue, i.e., from position j to position $j - 1$. In case of FIFO scheduling, v_j^0 would equal 1. For HOL-PJ priority scheduling, this is not necessarily the case, since new class 1 packets can arrive while the tagged packet is still waiting in the second queue and these class 1 packets have to be served before the tagged packet. Moreover, we have to take into account the possibility that during the waiting time of our packet, the contents of the second queue (including the tagged packet) jumps to the first one. This complicates the analysis even more than in the case of a static priority scheduling discipline.

Initially, we substitute $[u_{T,k} - 1]^+ + \hat{f}_{2,k}$ by the random variable B to simplify the analysis. Introducing pgf's in equation (17) yields

$$\begin{aligned}
D_2(z) &\triangleq \mathbf{E} [z^{d_2}] \\
&= \mathbf{E} \left[z^{\sum_{j=1}^B v_j^0 + 1} \right] \\
&= z \left(\mathbf{E}_B \left[\mathbf{E}_{v_j^0} \left[z^{\sum_{j=1}^B v_j^0} \{\text{jump before } v_B^0\} \right] \right. \right. \\
&\quad + \mathbf{E}_{v_j^0} \left[z^{\sum_{j=1}^B v_j^0} \{\text{no jump before } v_B^0, \text{ jump during } v_B^0\} \right] \\
&\quad + \mathbf{E}_{v_j^0} \left[z^{\sum_{j=1}^B v_j^0} \{\text{no jump before } v_B^0, \text{ no jump during } v_B^0, \text{ jump during } v_{B-1}^0\} \right] \\
&\quad + \dots + \\
&\quad + \mathbf{E}_{v_j^0} \left[z^{\sum_{j=1}^B v_j^0} \{\text{no jump before } v_B^0, \text{ no jump during } v_B^0, \text{ no jump during } v_{B-1}^0, \dots, \right. \\
&\quad \left. \text{no jump during } v_2^0, \text{ jump during } v_1^0\} \right] \\
&\quad \left. + \mathbf{E}_{v_j^0} \left[z^{\sum_{j=1}^B v_j^0} \{\text{no jump before } v_B^0, \text{ no jump during } v_B^0, \dots, \text{ no jump during } v_1^0\} \right] \right] \Big), \tag{18}
\end{aligned}$$

with $\mathbf{E}[X \{Y\}] \triangleq \mathbf{E}[X|Y] \text{Prob}[Y]$ and where the jump *before* v_B^0 means a jump during the arrival slot of the tagged packet. This last expression is found by conditioning on if and when a jump occurs during the delay of the tagged packet. For v_j^0 , three cases can be specified: the tagged packet is still in queue 2 at the beginning of v_j^0 and there is no jump of the contents of queue 2 during v_j^0 , the tagged packet is still in queue 2 at the beginning of v_j^0 but a jump

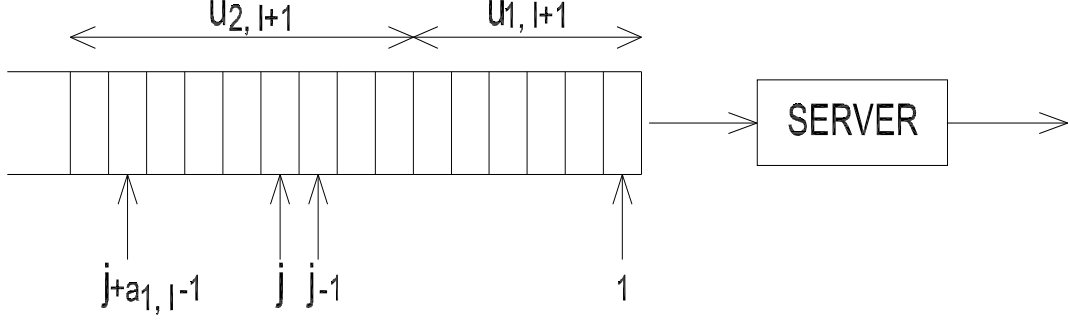


Figure 1: A logically equal queueing system

occurs during v_j^0 , and there was already a jump before v_j^0 and the tagged packet is thus already in queue 1. Let us define some partial pgfs:

$$\begin{aligned}
V_0(\beta, z) &\triangleq \mathbb{E} \left[z^{v_j^0} \{ \text{tagged packet stays in queue 2 during } v_j^0 \} \right], \\
V_1(\beta, z) &\triangleq \mathbb{E} \left[z^{v_j^0} \{ \text{tagged packet jumps from queue 2 to queue 1 during } v_j^0 \} \right], \\
V_2(\beta, z) &\triangleq \mathbb{E} \left[z^{v_j^0} \{ \text{tagged packet already in queue 1} \} \right].
\end{aligned}$$

When we express (18) in terms of these partial generating functions, we obtain

$$\begin{aligned}
D_2(z) &= z \mathbb{E}_B \left[\beta (V_2(\beta, z))^B + (1 - \beta) (V_1(\beta, z) V_2(\beta, z))^{B-1} + V_0(\beta, z) V_1(\beta, z) V_2(\beta, z)^{B-2} \right. \\
&\quad \left. + \dots + (V_0(\beta, z))^{B-1} V_1(\beta, z) + (V_0(\beta, z))^B \right] \\
&= z \left[\beta B (V_2(\beta, z)) + (1 - \beta) \left(V_1(\beta, z) \frac{B (V_2(\beta, z)) - B (V_0(\beta, z))}{V_2(\beta, z) - V_0(\beta, z)} + B (V_0(\beta, z)) \right) \right] \quad (19)
\end{aligned}$$

where $B(z) \triangleq \mathbb{E} [z^B]$.

We now compute the partial generating functions $V_0(\beta, z)$, $V_1(\beta, z)$ and $V_2(\beta, z)$. First we note that $V_2(\beta, z)$ equals z . Indeed, since in this case, the tagged packet is already in queue 1, no new arriving packets can be stored in front of it, and it thus takes 1 slot to move 1 position ahead in the queue, thus $v_j^0 = 1$. Assume that the tagged packet is stored in the j -th position in the queue at the beginning of the l -th slot ($0 < j < [u_{T,k} - 1]^+ + \hat{f}_{2,k}$), the tagged packet is still in queue 2 and a packet is served during slot l . If no class 1 packets arrive during slot l ,

v_j^0 will equal 1. But if $a_{1,l} (> 0)$ class 1 packets arrive during this slot, the tagged packet will move back to position $j + a_{1,l} - 1$ at the beginning of slot $l + 1$ in the queue, since these class 1 packets have to be served before all class 2 packets (currently in the system), and thus before the tagged one (Figure 1). Since these class 2 packets possibly jump *at the end* of slot l , this observation is independent from a possible jump. Let us define $v_{j,i}^1$ as the number of slots it takes for the tagged packet to move from position i to position $i - 1$. Then, it is clear that v_j^0 can be calculated as follows,

$$v_j^0 = \sum_{i=0}^{a_{1,l}-1} v_{j,j+i}^1 + 1. \quad (20)$$

It is obvious that all $v_{j,i}^1$ form a set of mutually independent random variables since they depend on the number of class 1 packet arrivals during different slots. For $V_0(\beta, z)$, we obtain

$$V_0(\beta, z) = (1 - \beta)zA_1(V_0(\beta, z)). \quad (21)$$

This expression can be explained as follows: since there are no jumps during the complete v_j^0 , there are no jumps during the various $v_{j,i}^1$'s, thus all the $v_{j,i}^1$'s have the same pgf $V_0(\beta, z)$. For the calculation of $V_1(\beta, z)$, we start with the same equation (20). During v_j^0 , there now occurs a jump. Introducing pgf's in this case, yields

$$V_1(\beta, z) = (1 - \beta)zV_1(\beta, z)\frac{A_1(z) - A_1(V_0(\beta, z))}{z - V_0(\beta, z)} + \beta zA_1(z).$$

This expression can be explained as follows: either there is no jump during the l -th slot (with probability $1 - \beta$) and there thus occurs a jump during at least one of the $v_{j,i}^1$'s, or either there is a jump during the l -th slot, and from that moment on, the tagged packet is in queue 1, and the $v_{j,i}^j$'s all equal 1. This equation can be simplified and finally, we obtain (using equation (21))

$$V_1(\beta, z) = \frac{\beta A_1(z)(z - V_0(\beta, z))}{1 - (1 - \beta)A_1(z)}. \quad (22)$$

Going back now to expression (19), the only function we have yet to calculate is $B(z)$. When we use the definition of $B = [u_{T,k} - 1]^+ + \hat{f}_{2,k}$, we get

$$B(z) = F_2(z) \frac{U_T(z) + (z-1)U_T(0)}{z}, \quad (23)$$

with $F_2(z) \triangleq \mathbb{E}[z^{f_{2,k}}]$. The pgf of $f_{2,k}$ has been calculated in [2]:

$$F_2(z) = \frac{A_T(z) - A_1(z)}{\lambda_2(z-1)}.$$

Substitution of this equation, equations (22)-(23) and equation (8) in equation (19), finally gives

$$\begin{aligned} D_2(z) &= \frac{\beta(1-\lambda_T)}{\lambda_2} \frac{z(A_T(z) - A_1(z))}{(z - A_T(z))(1 - (1-\beta)A_1(z))} \\ &+ \frac{(1-\beta)(1-\lambda_T)}{\lambda_2} \frac{z(A_T(V_0(\beta, z)) - A_1(V_0(\beta, z)))(1 - A_1(z))}{(V_0(\beta, z) - A_T(V_0(\beta, z)))(1 - (1-\beta)A_1(z))}, \end{aligned} \quad (24)$$

with $V_0(\beta, z)$ implicitly given as in (21).

Note that the pgf of the delay of an arbitrary (class 1 or class 2) packet equals

$$D(z) = \frac{\lambda_1}{\lambda_T} D_1(z) + \frac{\lambda_2}{\lambda_T} D_2(z), \quad (25)$$

since the probability that an arriving packet is of class j equals $\frac{\lambda_j}{\lambda_T}$ ($j = 1, 2$).

Finally, the same expressions for $D_1(z)$ and $D_2(z)$ as in [15] can be found by substituting $\beta = 0$ in (16) and (24), and by using l'Hopital's rule, yielding

$$\begin{aligned} D_1(z) &= \frac{1 - \lambda_1}{\lambda_1} \frac{z(A_1(z) - 1)}{z - A_1(z)}, \\ D_2(z) &= \frac{1 - \lambda_T}{\lambda_2} \frac{z(A_T(V_0(0, z)) - A_1(V_0(0, z)))}{V_0(0, z) - A_T(V_0(0, z))}, \end{aligned} \quad (26)$$

with $V(0, z) = zA_1(V(0, z))$, when $\beta = 0$.

5 Calculation of the moments

In this section, we give expressions for the mean values of the studied stochastic variables. To make the expressions more readable, we define λ_{11} and λ_{TT} as

$$\lambda_{11} \triangleq \left. \frac{\partial^2 A(z_1, z_2)}{\partial^2 z_1} \right|_{z_1=z_2=1} \quad \text{and} \quad \lambda_{TT} \triangleq \left. \frac{\partial^2 A_T(z)}{\partial z^2} \right|_{z=1}$$

respectively. By taking the first derivative of the respective pgf's for $z = 1$, we obtain expressions for the mean values of the total system contents and the system contents of both queues:

$$\mathbb{E}[u_T] = \lambda_T + \frac{\lambda_{TT}}{2(1 - \lambda_T)}, \quad (27)$$

$$\mathbb{E}[u_1] = 1 + \lambda_2 - \frac{1 - \lambda_1}{\beta} + \frac{\lambda_{TT}}{2(1 - \lambda_T)} - \frac{(1 - \lambda_T)(1 - \beta)A_T(Y(\beta, 1))}{Y(\beta, 1) - A_T(Y(\beta, 1))}, \quad (28)$$

$$\mathbb{E}[u_2] = \frac{(1 - \beta)\lambda_2}{\beta} + \frac{(1 - \lambda_T)(1 - \beta)(Y(\beta, 1) - (1 - \beta)A_T(Y(\beta, 1)))}{\beta(Y(\beta, 1) - A_T(Y(\beta, 1)))}. \quad (29)$$

It is easily verified that these three equations satisfy $\mathbb{E}[u_T] = \mathbb{E}[u_1] + \mathbb{E}[u_2]$. The expressions for the mean values of the packet delay are respectively

$$\mathbb{E}[d_1] = 1 + \lambda_2 - \frac{1 - \lambda_1}{\beta} + \frac{\lambda_{TT}\lambda_1 + \lambda_{11} - \lambda_{11}\lambda_T}{2(1 - \lambda_T)\lambda_1} - \frac{(1 - \lambda_T)A_T(Y(\beta, 1))}{Y(\beta, 1) - A_T(Y(\beta, 1))} \quad (30)$$

for a class 1 packet,

$$\begin{aligned} \mathbb{E}[d_2] &= 1 - \lambda_1 + \frac{\lambda_1}{\beta} + \frac{(1 - \lambda_1)\lambda_{TT} - (1 - \lambda_T)\lambda_{11}}{2(1 - \lambda_T)\lambda_2} \\ &\quad - \frac{(1 - \beta)(1 - \lambda_T)(A_T(V_0(\beta, 1)) - A_1(V_0(\beta, 1)))\lambda_1}{\lambda_2(V_0(\beta, 1) - A_T(V_0(\beta, 1)))\beta} \end{aligned} \quad (31)$$

for a class 2 packet and

$$\mathbb{E}[d] = \frac{\lambda_1}{\lambda_T}\mathbb{E}[d_1] + \frac{\lambda_2}{\lambda_T}\mathbb{E}[d_2] \quad (32)$$

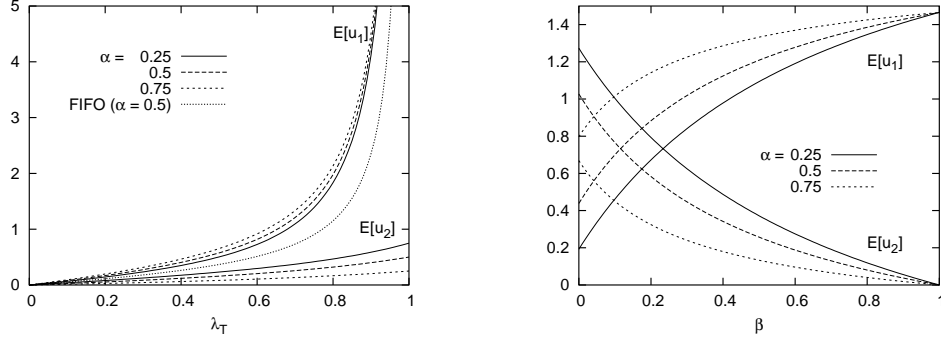
for an arbitrary packet. Once the values of $Y(\beta, 1)$ and $V_0(\beta, 1)$, which can be easily obtained numerically, are known, explicit expressions for all the moments can be calculated. Furthermore, notice that $E[u_j] = \lambda_j E[d_j]$ ($j = 1, 2$) does *not* hold, as one would, at first, expect according to Little's law. The reason for this is that in the calculation of the system contents, packets of the second queue jump to the first one and from that moment on, they are treated as part of the system contents of queue 1. This is of course not the case in the calculation of the packet delay of a class 2 packet. So basically, Little's law does not hold with respect to each queue separately, because the system contents is analyzed on a "queue"-basis, while the packet delay is analyzed on "packet"-basis. However, Little's law does hold for the whole system. Indeed, taking into account that $Y(\beta, 1) = V_0(\beta, 1)$, which follows directly from the definitions of $Y(\beta, z)$ and $V_0(\beta, z)$, it can be easily verified that $E[u_T] = \lambda_T E[d]$.

Expressions of higher and cross-moments can also be obtained in the same way as for the mean values (i.e., by taking higher order derivatives of the respective pgf's), but are omitted because of their size. Variances are however illustrated in figures in the next section.

6 Numerical examples & application

6.1 Numerical examples

In this section, the results obtained in the former sections, are applied to an output-queueing switch. This output-queueing switch has N inlets and N outlets and we assume two types of traffic: traffic of class 1 is delay-sensitive (for instance, voice) and traffic of class 2 is assumed to be delay-insensitive (for instance, data). The packet arrivals on the inlets are generated by identical and independent Bernoulli processes with arrival rate λ_T . An arriving packet is assumed to be of class j with probability λ_j/λ_T ($\lambda_1 + \lambda_2 = \lambda_T$). The incoming packets are then routed to the output queue corresponding to their destination, in an independent and uniform way. Therefore, the output queues all behave identically and we can concentrate on the analysis of one output queue. The numbers of arrivals of both types of packets to an output queue in one slot are



a. versus the total arrival rate for $\beta = 0.5$ b. versus β for $\lambda_T = 0.7$

Figure 2: Mean value of system contents

generated according to a two-dimensional binomial process, which is fully characterized by the following joint pgf

$$A(z_1, z_2) = \left(1 - \frac{\lambda_1}{N}(1 - z_1) - \frac{\lambda_2}{N}(1 - z_2)\right)^N. \quad (33)$$

Obviously, the numbers of class 1 and class 2 arrivals at an output queue during a slot are correlated. When m class 1 packets arrive at the tagged queue during a slot ($0 \leq m \leq N$), the maximum number of class 2 arrivals during the same slot is limited by $N - m$ (because there are only N inlets). At each output, two logical queues are present. Class 1 packets arrive to the first queue and class 2 packets arrive to the second queue. In each slot, the contents of queue 2 jumps to the first queue with probability β . So the results obtained in the former sections can be used to study the performance of this ATM switch.

We will now investigate the effect of the dynamic priority scheduling discipline on some performance measures, such as the mean value and the variance of system contents and packet delay. A static priority scheduling has been investigated in the past [15]. A static priority queue is a special case of the dynamic priority queue we have analyzed, namely without priority jumps, i.e., with $\beta = 0$. In the figures in this section, we will concentrate on the comparison between queues with dynamic priority scheduling, static priority scheduling and FIFO scheduling. In the remainder of this section, we assume a 16x16-switch ($N = 16$). Further, we define α as the fraction of class 1 arrivals in the overall traffic mix (i.e., $\alpha = \lambda_1/\lambda_T$).

In Figure 2a., the mean value of the system contents of queue 1 and queue 2 is shown for $\beta = 0.5$, as a function of the total arrival rate, when $\alpha = 0.25, 0.5$ and 0.75 respectively. In order to compare the dynamic priority scheduling discipline with the FIFO scheduling discipline, we have also applied FIFO scheduling on a joint queue in which the packets of both classes are mixed up (according to their arrivals). We have plotted the mean system contents of one class of packets, since for $\alpha = 0.5$, the mean system contents of class 1 packets equals the mean system contents of class 2 packets. We can easily see that the mean system contents of queue 1 is larger for the dynamic priority scheduling than for the FIFO scheduling. For the mean system contents of queue 2, the opposite holds. This can be explained as follows: packets of queue 1 have priority over packets of queue 2. So without priority jumps, the second queue would build up as long as there are packets in queue 1. Because of the priority jumps, the total contents of queue 2 jumps with a probability $\beta = 0.5$ to queue 1. So, on average, the contents of queue 2 jumps to queue 1 once every two slots, leaving the second queue totally empty. Therefore, the mean system contents of queue 1 is larger than the mean system contents of queue 2. The higher the arrival rate, the more accumulation in queue 2, but also the more drastic the growth of the mean system contents of queue 1. The figure also shows that the mean system contents of the first queue increases with α . This is logical, since bigger α means a higher fraction of high-priority packets in the arrival stream. The opposite holds for the second queue.

Figure 2b. shows the mean value of the system contents of queue 1 and queue 2 for $\lambda_T = 0.7$, as a function of β , when $\alpha = 0.25, 0.5$ and 0.75 . The influence of β is quite obvious: larger β means more jumps (on average), resulting into higher mean system contents of queue 1 and smaller mean system contents of queue 2. The figure also shows that the two curves for $\alpha = 0.25$ and the two curves for $\alpha = 0.5$ intersect each other for certain values of β , meaning that from those β -values on, the mean system contents of queue 1 is larger than the mean system contents of queue 2 for the respective fractions of class 1 packets in the arrival stream. For $\alpha = 0.75$, the mean system contents of queue 1 is always larger than the mean system contents of queue 2 (when $\lambda_T = 0.7$). When $\beta = 1$, we can easily see that the mean system contents of queue 2

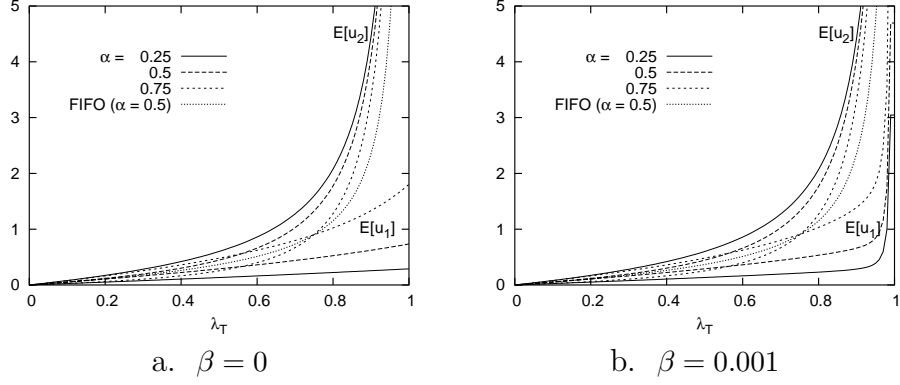


Figure 3: Mean value of system contents versus the total arrival rate

becomes 0. In each slot, the newly arriving packets of class 2 jump to the first queue immediately, leaving the second queue empty *in each slot*.

Figure 3 shows that $\beta = 0$ is really a special case. When $\beta = 0$, the mean system contents of queue 1 stays finite when λ_T approaches 1, while the mean system contents of queue 2 approaches infinity. When $\beta = 0.001$, the mean system contents of queue 1 does not stay finite when λ_T approaches 1. However, in this case the mean system contents of queue 2 stays finite. Indeed, since $1 - \lambda_T$ does not appear in the denominator of expression (29), it can be easily seen that the mean system contents of queue 2 stays finite for λ_T approaching 1 and $\beta > 0$. However, for small β , the mean system contents of queue 2 is very high (that is why we can not see it in the figure). It is due to the jumping mechanism ($\beta > 0$) that the mean system contents of queue 1 does not and the mean system contents of queue 2 does stay finite: as long as no jumps occur, both queues build up. The packets of queue 2 are being served, meaning that the mean system contents of queue 2 gets larger. When a jump occurs, all contents of queue 2 jumps to queue 1, leaving the second queue empty. So, sooner or later, there will be a jump, meaning that the system contents of queue 2 stays finite and the system contents of queue 1 approaches infinity when λ_T approaches 1.

In Figure 4, the variance of the system contents of both classes is shown as a function of the total arrival rate and β respectively (with the same characteristics as in Figure 2). Similar conclusions as for the mean system contents can be derived.

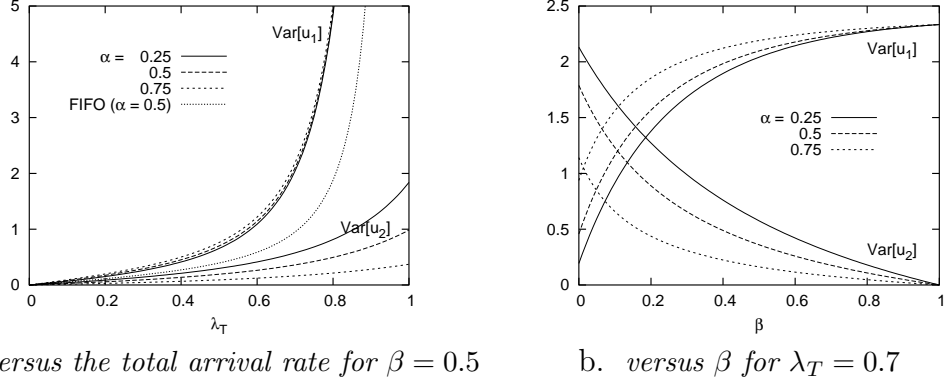


Figure 4: Variance of system contents

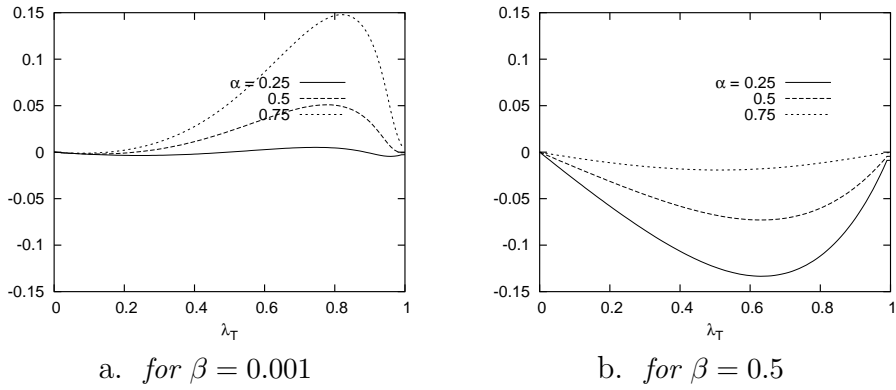


Figure 5: Correlation of system contents versus the total arrival rate

The next performance measure which we will discuss, is the correlation coefficient $\rho_{u_1 u_2}$. This performance measure quantifies the correlation between the number of packets in queue 1 and queue 2 in the system during a random slot. On the basis of four figures, we will observe this correlation and explain these observations. First, Figure 5a. shows the correlation coefficient for $\beta = 0.001$, as a function of the total arrival rate for $\alpha = 0.25, 0.5$ and 0.75 . We see that $\rho_{u_1 u_2}$ increases when the fraction of class 1 packets increases (for a given total load). This can be easily understood by the priority scheduling and by the fact that there are few jumps (on average, 1 jump in 1000 slots): the influence of packets in queue 1 on the system contents of queue 2 will become more important, when the fraction of class 1 packets increases. A second observation is that $\rho_{u_1 u_2}$ is slightly negative when the total load is small, but becomes positive when the total load is large. The reason for this are two counteracting mechanisms. The first one is the switch structure: when more class 1 packets arrive at the switch, there will be fewer class 2 packets

arriving at the same time (since the amount of inlets is limited), and vice versa. This negative correlation between packet arrivals of the two priority classes during a slot shows for small values of λ_T . For these parameter values, there is virtually no queuing and the buffer behavior is mainly determined by the number of arrivals during a single slot. The second influence is priority scheduling. As λ_T further increases, more packets are being queued, and the presence of high priority packets starts to hinder the transmission of low priority packets, thereby leading to a positive correlation between u_1 and u_2 . Finally, when λ_T approaches 1, the contents in queue 1 approaches infinity, due to the system becoming unstable. As a result, $\rho_{u_1 u_2}$ approaches 0 (since the contents in queue 2 stays finite when the jumping probability $\beta > 0$).

In Figure 5b., the correlation coefficient is shown for $\beta = 0.5$, as a function of the total arrival rate for $\alpha = 0.25, 0.5$ and 0.75 . In this case, the contents of queue 2, on average, jumps to queue 1 once every two slots, leaving the second queue totally empty. First, we see that $\rho_{u_1 u_2}$ becomes more negative when the fraction of *class 2* packets increases (for a given total load). This can be understood by the *dynamic* priority scheduling (or the jumping mechanism): the influence of class 2 packets on class 1 packets will become more important, when the fraction of class 2 packets increases. The second observation is that $\rho_{u_1 u_2}$ is not only negative when the total load is small, but stays negative when the total load is larger. Two main counteracting mechanisms are at work: first, as long as no jumps occur, both queues build up, resulting in a positive correlation coefficient between the contents in both queues. When a jump occurs, the contents of queue 2 jumps to queue 1, leaving the second queue empty. So when queue 2 is empty, there is a reasonable chance that a jump occurred in the near past and thus that the contents in queue 1 is reasonably high. Or in other words, when the contents in the first queue is large, there is a high probability that a jump occurred a few slots ago, and thus that the contents in queue 2 is low, resulting in a negative (overall) correlation coefficient. Finally, when λ_T approaches 1, the total system contents (and the number of packets in queue 1) approaches infinity, due to the system becoming unstable and as a result, $\rho_{u_1 u_2}$ approaches 0.

Figure 6a. shows the correlation coefficient for $\lambda_T = 0.1$, as a function of β for $\alpha = 0.25, 0.5$

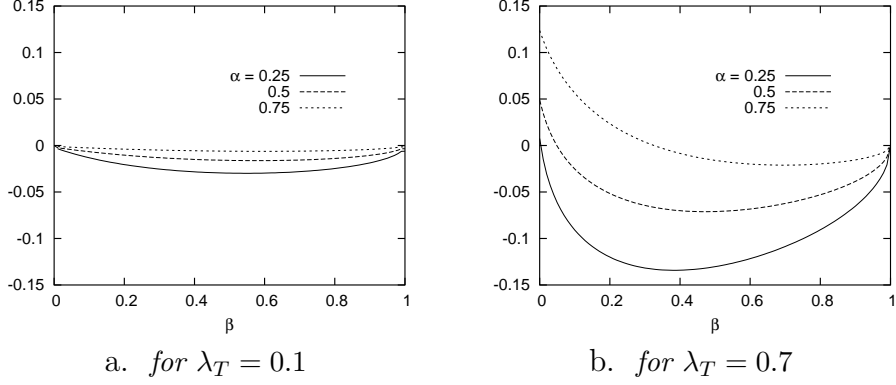
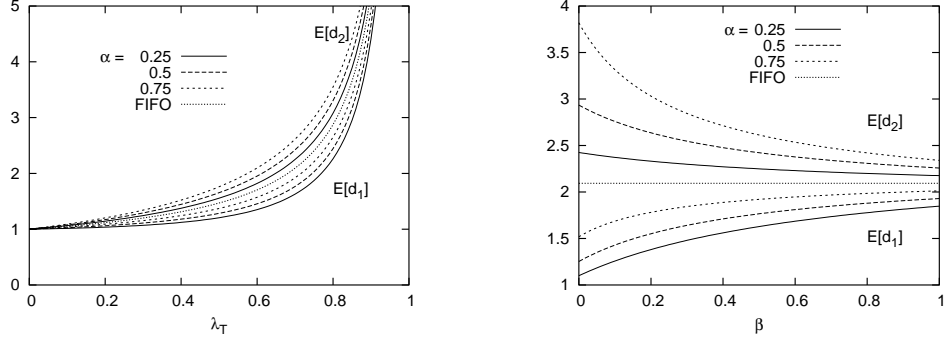


Figure 6: Correlation of system contents versus β

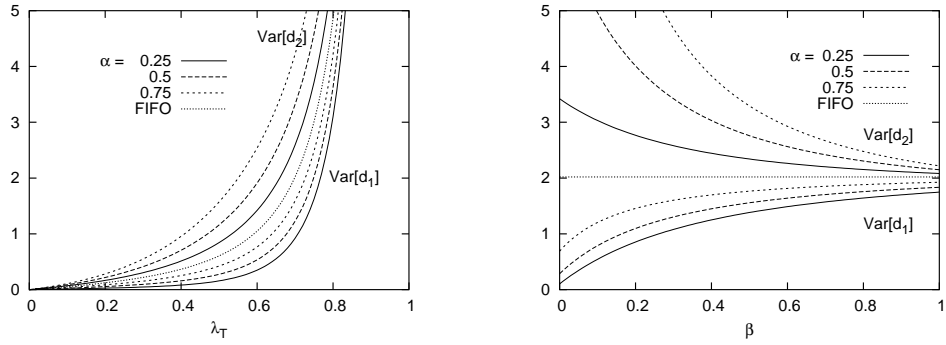
and 0.75. The first observation is again that $\rho_{u_1u_2}$ becomes more negative when the fraction of class 2 packets increases (for a given β): the influence of packets in queue 2 on packets in queue 1 is larger when the fraction of class 2 packets increases, due to the jumping mechanism. Secondly, we see that $\rho_{u_1u_2}$ is slightly negative when β is small, and stays negative when β is large. We have seen before that when λ_T is small, there is virtually no queueing, meaning that the buffer behavior is mainly determined by the arrival process. For small values of β (on average, few jumps), the jumping mechanism does not play a big role. But as β further increases, more jumps occur (on average). And as described above, this jumping mechanism leads to a negative correlation coefficient between the system contents of both queues. Finally, when β approaches 1, the probability that the second queue is empty, becomes very high (this probability is one for $\beta = 1$). As a result, having an empty second queue becomes so common that it does not tell much about the contents of the first queue, and thus $\rho_{u_1u_2}$ approaches 0 as β goes to 1.

Finally, in Figure 6b., the correlation coefficient is shown for $\lambda_T = 0.7$, as a function of β for $\alpha = 0.25, 0.5$ and 0.75 . $\rho_{u_1u_2}$ is slightly positive when β is small, but becomes negative when β is large. The reason for this is the jumping mechanism. We notice that there is a lot of queueing when $\lambda_T = 0.7$. For very small values of β , the jumping mechanism hardly plays a role. In this case, the presence of high priority packets hinders the transmission of low priority packets, thereby leading to a positive correlation between u_1 and u_2 . But as β further increases, the jumping mechanism results in a negative correlation. Finally, as β goes to 1, this correlation coefficient goes to 0.



a. versus the total arrival rate for $\beta = 0.5$ b. versus β for $\lambda_T = 0.7$

Figure 7: Mean value of packet delays



a. versus the total arrival rate for $\beta = 0.5$ b. versus β for $\lambda_T = 0.7$

Figure 8: Variance of packet delays

Figure 7a. shows the mean packet delay of both classes when $\beta = 0.5$ as a function of the total load for $\alpha = 0.25, 0.5$ and 0.75 respectively. In order to compare with FIFO scheduling, we have also shown the mean value of the packet delay in that case. The packet delay is then the same for class 1 and class 2 packets (independent of α), and can thus be calculated as if there is only one class arriving according to an arrival process with pgf $A(z, z)$ (see [3]). The influence of the priority scheduling is quite obvious: the mean packet delay of class 1 is smaller for the dynamic priority scheduling than for the FIFO scheduling. For the mean packet delay of class 2, the opposite holds. The reason is clear: the packets of class 1 have priority over those of class 2. However, the influence of dynamic priority scheduling is limited. The mean delay of class 1 packets reduces only moderately in comparison with FIFO scheduling, while the price to pay, a higher mean packet delay for class 2 packets, is rather small. Also note that it follows from this figure that increasing the fraction of high-priority packets in the overall traffic mix

(i.e., increasing α), increases the mean delay of high- and low-priority packets.

In Figure 7b., the mean packet delay of both classes is shown for $\lambda_T = 0.7$, as a function of β , when $\alpha = 0.25, 0.5$ and 0.75 . The larger β (and thus the more jumps), the lower the negative effect from the priority scheduling on the low-priority packet delay. The price to pay is a higher mean delay for high-priority packets. So we conclude that the dynamic priority scheduling does what it is designed for: lowering the (mean) delay of the high-priority packets, but, in contrast with the static priority scheduling ($\beta = 0$), taking into consideration the (mean) delay of the low-priority packets. The parameter β can be chosen depending on the delay guarantees for both types of traffic. A low β will highly favour the high-priority traffic, while choosing β higher will give the high-priority traffic only a small delay reduction (compared to the FIFO scheduling). We can derive similar conclusions with respect to delay jitter from Figure 8, which show the variance of the delay of both classes for $\alpha = 0.25, 0.5$ and 0.75 versus λ_T (Figure 8a.) and β (Figure 8b.) respectively, when $\beta = 0.5$ and $\lambda_T = 0.7$ respectively.

Finally, we say something about the mean value and the variance of the delay of an arbitrary packet. For a given total load, the mean delay of an arbitrary packet is the same for all α and all β , and equals the mean packet delay in the case of a FIFO scheduling discipline. This can be easily understood by Little's law. As we have seen in Section 3, the probability generating function of the total system contents equals the pgf of the system contents of a queue with a FCFS-discipline and with one class (with arrivals determined by $A_T(z)$), meaning that the mean values of the system contents are also identical. Now, according to Little's law, the mean value of the total system contents equals the total arrival rate multiplied by the mean value of the delay of an arbitrary packet, and thus it follows that the mean delay of an arbitrary packet is identical for all α and β , and is equal to the mean delay in an identical queue with a FIFO scheduling discipline.

In Figure 9, the variance of the delay of an arbitrary packet is shown as a function of β with $\lambda_T = 0.7$, for $\alpha = 0.25, 0.5$ and 0.75 . From this figure, we conclude that the studied HOL-PJ priority discipline with $\beta = 1$ results in an equal variance of the delay of an arbitrary packet

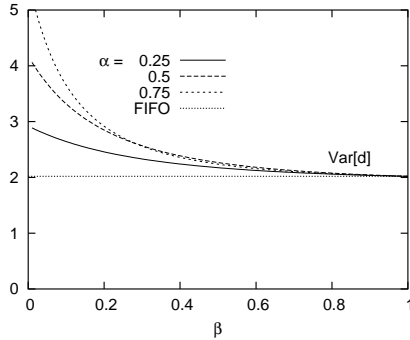


Figure 9: Variance of delay of an arbitrary packet versus β for $\lambda_T = 0.7$

as for a FIFO scheduling discipline. Letting $\beta < 1$ however, results in a higher variance of the delay of an arbitrary packet. Indeed, lowering β means that the delay of class 1 packets lowers, while the delay of class 2 packets is bigger, thus resulting in a higher variance of the delay of an arbitrary packet (since the mean delay of an arbitrary packet is independent of β).

6.2 Application

In this final subsection, we show that the performance of the output-queueing switch described in the previous subsection, for our introduced HOL-PJ scheme hardly differs from the performance for the original HOL-PJ scheme of [14], with regard to the mean packet delay of both classes and of an arbitrary packet in particular.

To be able to compare both schemes, we have simulated the original one. As explained in the introduction, this scheme makes use of a limit L , which is set on the maximum queueing delay for packets in the low-priority queue and which makes packets, whose delay equals this limit, jump to the high-priority queue. To make a valid comparison between both schemes, β should be properly defined (as a function of L). Let us observe a tagged low-priority packet. Starting from the first slot after its arrival slot, the number of slots until the packet jumps is (shifted) geometrically distributed with parameter $1 - \beta$ (since it stays in queue 2 during a slot with probability $1 - \beta$). The corresponding average number of slots then equals $1/\beta$ (provided it is not served before jumping). Since L equals the number of slots a class 2 packets stays in queue 2 in the original system, β is defined as $1/L$. We now present some figures in which we compare

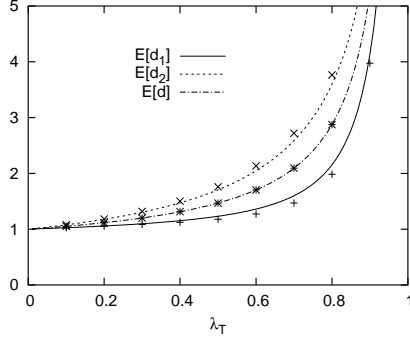


Figure 10: Mean value of packet delays versus the total arrival rate for $L = 4$ ($\beta = 0.25$) and $\alpha = 0.5$

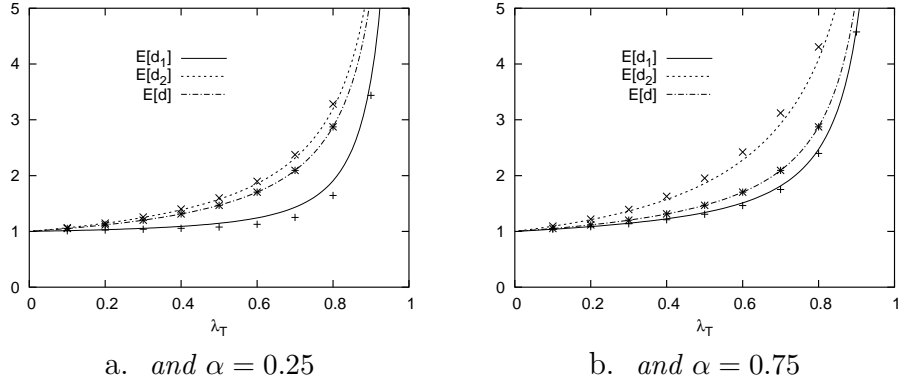


Figure 11: Mean value of packet delays versus the total arrival rate for $L = 4$ ($\beta = 0.25$)

both schemes (the simulation results of the original scheme are indicated by marks).

Figure 10 shows the mean packet delay of both classes and the mean packet delay of an arbitrary packet when $L = 4$ (and thus $\beta = 0.25$), as a function of the total load, for $\alpha = 0.5$. It can be seen from the figure that the performance of the output-queueing switch for both schemes hardly differs, with respect to the mean packet delay of both classes, and that the mean packet delay of an arbitrary packet is the same for both schemes. We also notice that the difference between the mean packet delay of class 1 for both schemes equals the difference between the mean packet delay of class 2 for both schemes. This can all be explained by Little's law and by noting that the mean total system contents is equal for both schemes (since they are both work-conserving and all packets have the same service time). So, $E[d]_s = E[d]_o$, and since

$$E[d] = \alpha E[d_1] + (1 - \alpha) E[d_2] \quad (34)$$

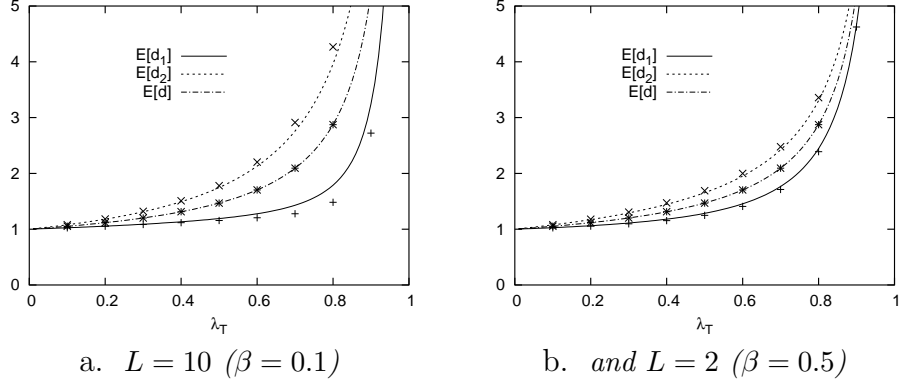


Figure 12: Mean value of packet delays versus the total arrival rate for $\alpha = 0.25$

for both schemes, we find

$$\alpha E[d_1]_s + (1 - \alpha) E[d_2]_s = \alpha E[d_1]_o + (1 - \alpha) E[d_2]_o,$$

which can be arranged as

$$E[d_1]_s - E[d_1]_o = \frac{1 - \alpha}{\alpha} (E[d_2]_o - E[d_2]_s). \quad (35)$$

So, when $\alpha = 0.5$,

$$E[d_1]_s - E[d_1]_o = E[d_2]_o - E[d_2]_s.$$

Formula (35) means that the difference between the mean packet delay of class 1 for both schemes is to the difference between the mean packet delay of class 2 for both schemes as $1 - \alpha$ is to α . This is confirmed by Figures 11a. and 11b., which show the mean packet delay of both classes and the mean packet delay of an arbitrary packet when $L = 4$ (and thus $\beta = 0.25$), as a function of the total load, for $\alpha = 0.25$ and $\alpha = 0.75$ respectively.

In Figures 12a. and 12b., we can see the mean packet delay of both classes and the mean packet delay of an arbitrary packet when $\alpha = 0.5$, as a function of the total load, for $L = 10$ ($\beta = 0.1$) and $L = 2$ ($\beta = 0.5$) respectively. The larger L , the more difference in performance between both schemes. This is due to the distribution of the number of slots until a packet

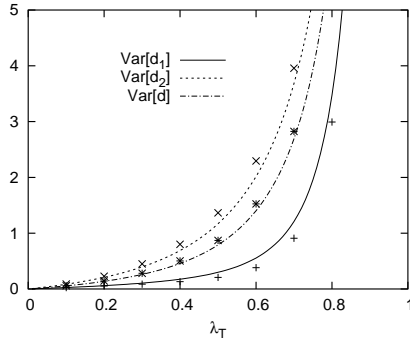


Figure 13: Variance of packet delays versus the total arrival rate for $L = 4$ ($\beta = 0.25$) and $\alpha = 0.5$

jumps. This number, as seen in a previous paragraph, is (shifted) geometrically distributed with mean L in the simplified scheme, while it is deterministic in the original HOL-PJ scheme. The variance of the first equals $L(L-1)$, while that of the second equals 0. So, the larger L , the larger the difference in variance. This explains the larger difference in performance between schemes for large L . Note the difference is still small for $L = 10$ though.

Finally, Figure 13 shows the variance of the delay of both classes and the variance of the delay of an arbitrary packet when $L = 4$ ($\beta = 0.25$), as a function of the total load, for $\alpha = 0.5$. We conclude from the figure that the difference in performance for both schemes - with respect to the variance of the delay - is limited.

7 Conclusions

In this paper, we have analyzed a queueing system with a newly introduced, simplified HOL-PJ priority scheduling discipline. A generating-function approach was adopted, which led to expressions of performance measures, such as mean and variance of the system contents in both queues and mean and variance of packet delay of both types of packets, that are easy to evaluate. The model includes possible correlation between the numbers of arrivals of the two traffic types during a slot. As a consequence, the results could be used to evaluate the performance of a dynamically prioritized output-queueing switch with Bernoulli arrivals. Finally, it is shown that simplifying the jumping mechanism hardly affects the original HOL-PJ scheme, while it is

easier to implement and analyze.

References

- [1] J.J. Bae and T. Suda. Survey of traffic control schemes and protocols in ATM networks. *ACM Transactions in Networking*, 2(5):508–519, 1994.
- [2] H. Bruneel and B.G. Kim. *Discrete-time models for communication systems including ATM*. Kluwer Academic Publishers, Boston, 1993.
- [3] H. Bruneel, B. Steyaert, E. Desmet, and G.H. Petit. An analytical technique for the derivation of the delay performance of ATM switches with multiserver output queues. *International Journal of Digital and Analog Communication Systems*, 5:193–201, 1992.
- [4] R. Chipalkatti, J.F. Kurose, and D. Towsley. Scheduling policies for real-time and nonreal-time traffic packet switching node. In *IEEE INFOCOM '89*, pages 774–783, 1989.
- [5] B.D. Choi, B.I. Choi, Y. Lee, and D.K. Sung. Priority queueing system with fixed length packet-train arrivals. *IEE Proceedings-Communications*, 145(2):331–341, 1998.
- [6] B.I. Choi and Y. Lee. Performance analysis of a dynamic priority queue for traffic control of bursty traffics in ATM networks. *IEE Proceedings-Communications*, 148(3):181–187, 2001.
- [7] D.I. Choi, B.D. Choi, and D.K. Sung. Performance analysis of priority leaky bucket scheme with queue-length-treshold scheduling policy. *IEE Proceedings-Communications*, 145(6), 1998.
- [8] S. Fratini. Analysis of a dynamic priority queue. *Commun. Statist.-Stochastic Models*, 6(3):415–444, 1990.
- [9] A. Gravey and G. Hebuterne. Mixing time and loss priorities in a single server queue. In *13th International Teletraffic Congress*, pages 147–152, 1991.
- [10] L. Kleinrock. *Queueing Systems*. Wiley, New York, 1975.

- [11] C. Knessl, D.I. Choi, and C. Tier. A dynamic priority queue model for simultaneous service of voice and data traffic. *SIAM Journal on Applied Mathematics*, 63(2):398–422, 2002.
- [12] J.T. Lee and Y.H. Kim. Performance analysis of a hybrid priority control scheme for input and output queueing ATM switches. In *Proceedings INFOCOM '98*, 1998.
- [13] Y. Lee and B.D. Choi. Queueing system with multiple delay and loss priorities for ATM networks. *Information Sciences*, 138(1-4):7–29, 2001.
- [14] Y. Lim and J.E. Kobza. Analysis of a delay-dependent priority discipline in an integrated multiclass traffic fast packet switch. *IEEE Transactions on Communications*, 38(5):659–685, 1990.
- [15] J. Walraevens, B. Steyaert, and H. Bruneel. Performance analysis of a single-server ATM queue with a priority scheduling. *Computers and Operations Research*, 30(12):1807–1829, 2003.