A CONSIDERATE PRIORITY QUEUEING SYSTEM WITH GUARANTEED POLICY FAIRNESS

IOANNIS STAVRAKAKIS

University of Vermont,
CS/EE Dept., Burlington, Vermont 05405, U.S.A.*

ABSTRACT A discrete-time queueing system supporting two classes of customers (packets of information) with different priorities is analyzed. Unlike the head of the line priority policy, the priority policy considered here provides for limited service to the low priority class, even in the presence of high priority customers (considerate system). At the same time, it guarantees that no low priority customer will be served before a previously arrived high priority one (guaranteed policy fairness). The proposed policy can be seen as a compromise between the head of the line priority policy and the classical gated / limited service priority policy. A general methodology is developed for the analysis of this policy.

I. INTRODUCTION

Discrete-time queueing models have been widely adopted for the analysis of packet communication networks, where packet processes are described by discrete time stochastic point processes [1]-[3]. Queueing systems are formulated in the networks due to the sharing of these resources by many users. The proliferation of services provided by today's networking systems and the prevailing trend to provide all services through a common integrated services network, have increased the complexity of the protocols governing the network operation. As a result, the queueing models adopted for the description of the operation of certain network elements have become more complex. The latter is due to the increased complexity in the description of both the packet arrival processes, generated by sources with widely different characteristics and the service policy, necessary for the accommodation of different requirements. Queueing systems with priorities are examples of systems with increased complexity due to the service policy. Such systems are formulated, for example, in the buffer of the transmitting end of a network link supporting packetized information with different delay requirements.

Research supported by the National Science Foundation under Grant NCR-9011962.

Priority queueing systems have been studied extensively in the past, [3]-[8]. A queue operating under two priority levels can be seen as consisted of two distinct queues. Let $\mathcal{H}-Q$ and $\mathcal{L}-Q$ denote the high and the low priority queues, respectively. The server switches between the queues in accordance with the service discipline. Under the Head Of the Line (HOL) policy, the server moves to $\mathcal{L}-Q$ only if $\mathcal{H}-Q$ is empty; it switches back to $\mathcal{H}-Q$ as soon as this queue becomes non-empty. As a result, this policy is inconsiderate to the low priority customers, which may suffer unfairly long delays.

A more considerate policy to the low priority customers is the gated / limited service policy. According to this policy, only the customers found in $\mathcal{H}-Q$ at the switching instant of the server to that queue, are served (gated service). Then, the server switches to $\mathcal{L}-Q$ and provides some limited service to that queue. Although this policy provides faster service to the high priority class, it is not completely fair to that class. The unfairness is due to the fact that high priority customers may be served after simultaneously (or at a later time) arrived low priority ones, even if only one customer limited service is provided to $\mathcal{L}-Q$ at each server visit. For instance, if $\mathcal{L}-Q$ is left empty upon switching to $\mathcal{H}-Q$ and both a low and a high priority customer arrive after the switching instant and before the completion of the gated service to $\mathcal{H}-Q$, then the low priority customer will be served before the high priority one, even if the latter had arrived before the former.

The queueing system considered in this paper (Section II) is a modified version of the gated / limited service policy. This policy guarantees fairness in the service, in the sense that no low priority customer will be served before a high priority one, which arrived at an earlier time (or simultaneously). In addition to this, the limited service to $\mathcal{L}-Q$ does not distinguish among low priority customers which arrived simultaneously. In the slotted, discrete-time environment

* Email: stavraka@uvm.edu

 CH3133-6/92/0000-2151 $3.00

considered in this paper, the previous is reflected by the provision of service to all low priority customer arrivals within the same time unit (slot), before the server leaves $\mathcal{L}-Q$.

The proposed queueing system is a discrete-time version of that formulated in [9] as a queueing model for a station in the DQDB (IEEE 802.6) Metropolitan Area Network [10]. Exact analysis of the continuous-time system in [9] and the one proposed here have not been carried out in the past. As described in [9], $\mathcal{L}-Q$ may be seen as the queue at a tagged DQDB station which is allowed to transmit only after all the reservations (high priority customers) made before have been served. The proposed priority service policy can be applied to establish fairness to a gated / limited type of service discipline and improve the performance of the high priority class.

II. THE QUEUEING SYSTEM

The queueing system considered in this paper is now described. Let $\mathcal{H}$ denote the class of packets which are favored (given priority) by the service policy of the system. Let $\mathcal{L}$ denote the class of packets which do not belong in class $\mathcal{H}$. To facilitate the description of the operation of the system and the adopted service policy, it may be assumed that new arrivals from a certain class join a specific queue (buffer) assigned to that class. As a result, two queues are formed, as shown in Fig. 1. Let $\mathcal{H}-Q$ and $\mathcal{L}-Q$ denote the high and the low priority queues, respectively.

Time is assumed to be slotted and the service time deterministic and equal to one slot. The packet arrival processes associated with the two classes are assumed to be mutually independent discrete-time arrival processes. For each priority class, the number of packet arrivals over a slot follows a general distribution. Packet arrivals over consecutive slots are assumed to be independent. A group of $\mathcal{L}$-packets ($\mathcal{H}$-packets) is defined to be the set of all $\mathcal{L}$-packets ($\mathcal{H}$-packets) arriving over the same slot. Events (packet arrivals and service completions) are assumed to occur at the slot boundaries.

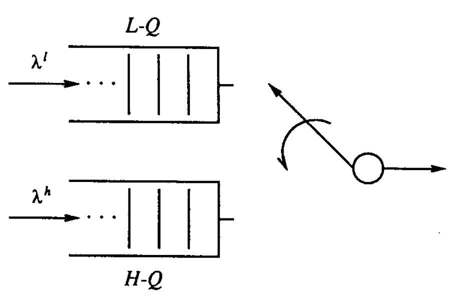

Fig. 1 The queueing system.

The service policy considered in this paper is a discrete-time version of that formulated in [9]. The server switching time between the queues is assumed to be zero. The service discipline is described by the following rules:

(a) The system is work conserving (*WC*). That is, the server is never idle when a packet is in the system and the discipline does not affect the amount of service time or the arrival time of any customer.

(b) When the system is empty the server is considered to be in a neutral position. If an $\mathcal{H}$-packet ($\mathcal{L}$-packet) arrives to a previously empty system, the server visits and starts serving $\mathcal{H}-Q$ ($\mathcal{L}-Q$) at the beginning of the next slot. If both $\mathcal{H}$-packets and $\mathcal{L}$-packets arrive to a previously empty system, the server starts serving $\mathcal{H}-Q$; then, it operates as the policy indicates.

(c) Packets within each queue are served according to a FIFO (First in-First Out) service policy.

(d) The server switches from $\mathcal{L}-Q$ to $\mathcal{H}-Q$ after serving the group of packets which contains the packet at the head of the $\mathcal{L}-Q$ (limited 1 group service).

(e) If upon switching to $\mathcal{H}-Q$, $\mathcal{L}-Q$ is left non-empty, then the server serves all $\mathcal{H}$-packets present in $\mathcal{H}-Q$ at the switching instant. Then it switches back to $\mathcal{L}-Q$.

(f) If upon switching to $\mathcal{H}-Q$, $\mathcal{L}-Q$ is left empty, then the server remains at $\mathcal{H}-Q$ and serves all the $\mathcal{H}$-packets which arrived prior to or over the same slot with the next $\mathcal{L}$-packet. Then, the server switches back to $\mathcal{L}-Q$.

It should be noted that the above described service policy guarantees that $\mathcal{H}$-packets will be served before any $\mathcal{L}$-packet which arrives over the same or at future time slots (guaranteed policy fairness). This is not the case with the classical gated / limited service policy, as discussed in Section I. The HOL priority has the fairness property, as defined before, but it is very inconsiderate to the $\mathcal{L}$-packets. The adopted policy guarantees that one group of $\mathcal{L}$-packets (if $\mathcal{L}-Q$ is non-empty) will be served after the gated service of the $\mathcal{H}-Q$ (limited 1-group service).

In view of the above discussion, it should be clear that the adopted service policy is a compromise between the classical gated and the HOL priority queueing disciplines.

III. ANALYSIS OF THE QUEUEING SYSTEM

III.1. The proposed general approach

In this section the performance of the adopted queueing discipline is evaluated. The analysis approach developed for this purpose is based on renewal and regenerative theory arguments. A similar approach has been applied in the past for the analysis of distributed queueing systems, [13]-[15], as they are formulated in random-access multi-user communication networks. This is the first time that such a methodology is applied for the analysis of priority queueing systems. The most difficult part in applying this methodology to the analysis of random-access multi-user protocols is related to the establishment of the system stability region and the conditions for the existence of a non-negative and finite solution to an infinite dimensional system of linear equations. This is trivially carried out in the case of priority queueing systems, as long as the service policy is non-preemptive and work-conserving (*NP–WC*). The supporting theory and the general methodology are presented in this sub-section.

Let the *equivalent* FIFO system be defined as the queueing system which has the same characteristics with the (priority) queueing system under consideration, but it operates under the FIFO service policy. Quantities associated with the equivalent FIFO system will be marked with the superscript FIFO. The following propositions present basic results from queueing theory for *WC* queueing systems, [5], [6], [11]. They are presented here in a discrete-time slotted context to be directly applied to the study of the queueing system described earlier. The continuous-time versions, if different at all, present only minor technical differences.

Proposition 1

The busy and idle period processes of a *WC* system and its *equivalent* FIFO systems are identical for all realizations. □

Let x be the random variable which describes the length (in slots, or discrete-time units) of the time interval between two consecutive instants when the *WC* system is empty. The following Lemma is obvious in view of Proposition 1.

Lemma 1

For a *WC* system $x=x^{FIFO}$ for all realizations and thus $X=X^{FIFO}$, where $X=E\{x\}$ and $E\{\cdot\}$ denotes the expectation operator. □

Let λ and $E\{s\}$ denote the arrival rate and the mean service time of the arrivals in the *equivalent* FIFO systems; let $\rho=\lambda E\{s\}$ denote the utilization of the system. Then, under the condition, [5],

$$\rho<1 \text{ and } E\{s^2\}<\infty \quad , \tag{1}$$

the queue is stable, the induced delay in the system is finite and ρ is equal to the fraction of time that the server is idle or the system is empty. The latter is also given by $1/X^{FIFO}$. Since the stability condition of a *WC* queueing system is not affected by the order of service, the following Lemma can be easily proved, in view of Proposition 1 and Lemma 1.

Lemma 2

For a *WC* queueing system, X is given by

$$X=\frac{1}{1-\rho}<\infty \quad , \text{ under the stability conditions (1)} \tag{2}$$

and the operation of the system induces renewal points of finite mean cycle length. □

Lemma 2 will be used for the establishment of the existence and the actual calculation of lower bounds on the delay associated with each of the priority classes of a *NP–WC* priority system. Then, upper bounds will be derived by using the Corollary to the next Theorem, under the conditions stated in the Theorem.

Theorem 1

Consider a *NP–WC* priority queueing system supporting K priority classes. Let λ^i denote the arrival rate of the i^{th} priority customers. Under the assumption that the customer service requirements do not depend on their priority class, [5], [6], [11],

$$\lambda D^{FIFO}=\sum_{i=1}^{K}\lambda^i D^i \quad , \quad \lambda=\sum_{i=1}^{K}\lambda^i \tag{3}$$

where D^i denotes the mean delay of the i^{th}-priority customer.

Let D^i_{lo}, $1\leq i\leq K$, denote a lower bound on the mean delay of the i^{th} priority customers in a priority queueing system, as described in Theorem 1. The following Corollary provides for an upper bound on D^i, $1\leq i\leq K$; its proof is evident in view of Theorem 1.

Corollary 1

An upper bound, D^i_{up}, on D^i, $1\leq i\leq K$, for the priority queueing system described in Theorem 1, is given by

$$D^i_{up}=\frac{1}{\lambda^i}\left[\lambda D^{FIFO}-\sum_{k=1,k\neq i}^{K}\lambda^k D^k_{lo}\right] . \quad \square \tag{4}$$

Note that the above theory is valid for a *NP–WC* queueing system supporting fixed length packets with different priorities. From now on, the customers will be considered to be fixed length packets whose service time is equal to 1 slot, independently of the priority class in which they belong. Packet arrivals associated with each priority class are assumed to form an independent and identically distributed process. These processes are assumed to be mutually independent.

Let $\{z_j\}_{j\geq1}$ denote the sequence of slot boundaries at which the queueing system is empty; $\{z_j\}_{j\geq1}$ is a renewal sequence with mean cycle length given by (2), under the stability conditions (1). Let $\{x_j\}_{j\geq 1}$ denote the sequence of the lengths of these cycles. Let c_j^i denote the cumulative delay of the i^{th} priority packets which arrived (were transmitted) over the j^{th} cycle. Under stability conditions, $\{c_j^i\}_{j\geq 0}$ is a regenerative process with respect to the renewal process $\{z_j\}_{j\geq 0}$ with $E\{c_j^i\}=C^i<\infty$. Thus, the mean delay of a i^{th} priority packet can be obtained from, [13], [14],

$$D^i=\frac{C^i}{\lambda^i X}=\frac{1-\rho}{\lambda^i}C^i \qquad (5)$$

To compute the expected value of the cumulative delay of the i^{th} priority packets, the specific priority discipline has to be taken into consideration. This is carried out in the sequel for the queueing system described in the previous section. The approach can potentially be applied to other two priority queueing systems. It turns out that the computation of C^i requires the solution of an infinite number of linear equations. A lower bound on C^i, C_{lo}^i, is obtained by solving a truncated, finite set of these equations; then, a lower bound on D^i, D_{lo}^i, is obtained by substituting C_{lo}^i in (5). Finally, Corollary 1 is invoked for the computation of an upper bound on D^i, D_{up}^i, by utilizing the lower bounds on D^i, for $0\leq i\leq K$. By considering a sufficiently large number of equations in the truncated version, arbitrarily close bounds can be obtained. The approach is illustrated and justified in the sequel, where lower bounds on the mean $\mathcal{L}$-packet and $\mathcal{H}$-packet delays are obtained for the system described in Section II.

III.2. Mean packet delay bounds for the 2-priority system

Consider the 2-priority *NP–WC* queueing system described in Section II. The deterministic packet service time (one slot) is adopted as the time unit. Packet arrivals and departures are declared at the end of the slots. The slot boundaries determine the discrete-time axis which is the time reference for all processes involved in the analysis of the system.

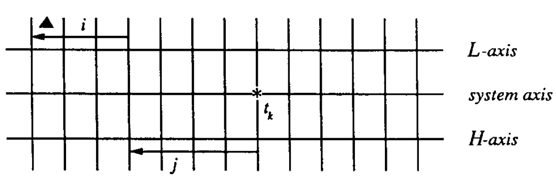

Fig. 2 Time axes for the definition of the state of the system.

For reasons which become clear later, the mean cycle length X, given by (2), is computed first. The developed approach will be directly applied to the calculation of C^h (C^l), where C^h (C^l) is the expected value of the cumulative $\mathcal{H}$-packet ($\mathcal{L}$-packet)) delay within a cycle.

Let $\{t_k\}_{k\geq 0}$ denote the sequence of time instants at which the service of the $\mathcal{H}$-packets found upon the k^{th} visit to the $\mathcal{H}-Q$ is completed. At a time instant t_k, let j_k, $0\leq j_k<\infty$, denote the length (in slots) of the unexamined interval on the $\mathcal{H}$-axis; that is, $\mathcal{H}$-packets which arrived over j_k have not been considered for service by t_k (Fig. 2). Let i_k+j_k, $0\leq i_k<\infty$, describe the distance from t_k of the group of the $\mathcal{L}$-packets which contains the packet at the head of the $\mathcal{L}-Q$ at time t_k. Let $\{r_k\}_{k\geq 0}$ be a stochastic process embedded at $\{t_k\}_{k\geq 0}$ with state space $S=\{(i,j) : 0\leq i<\infty, 0\leq j<\infty\}$, where i and j are the values of i_k and j_k at the current instant $t_k\in\{t_k\}_{k\geq 1}$. Since packet arrivals over consecutive slots are independent, it is easily established that $\{r_k\}_{k\geq 0}$ is a Markov chain embedded at $\{t_k\}_{k\geq 1}$. The following quantities are used in the analysis.

- $y(i,j)$: A random variable describing the length of the time interval (in slots) between some time instant t_k (as defined above) when the system is in state (i,j), and the first time in the future when the system becomes empty, $i\geq 0$, $j\geq 0$ and $i+j\neq 0$.
- $y(0,0)$: A random variable describing the length of the time interval between two consecutive instants when the system is empty; notice that $y(0,0)$ is the same as x, defined earlier.
- $Y(i,j)$: Expected value of $y(i,j)$, $i\geq 0$, $j\geq 0$.
- l : A random variable describing the number of $\mathcal{L}$-packets arrived over a slot. Let $g^l(k)$, $0\leq k\leq M^l<\infty$, and λ^l denote its probability mass function and its expected value, respectively.
- l_c : A random variable describing the number of $\mathcal{L}$-packets arrived over a slot, given a group of $\mathcal{L}$-packets has arrived over this slot. Let $g_c^l(k)$, $1\leq k\leq M^l$, and μ^l denote its probability mass function and its expected value, respectively.
- h_1 : A random variable describing the number of $\mathcal{H}$-packets arrived over a slot. Let $g^h(k)$, $0\leq k\leq M^h<\infty$, and λ^h denote its probability mass function and its expected value, respectively.
- h_c : A random variable describing the number of $\mathcal{H}$-packets arrived over a slot given that a group of $\mathcal{H}$-packets has arrived over this slot. Let $g_c^h(k)$, $1\leq k\leq M^h$, and μ^h denote its probability mass function and its expected

value, respectively.

h_k : A random variable describing the number of $\mathcal{H}$-packets arrived over k slots. Let $h(k,j)$, $0 \le j \le kM^h$, denote its probability mass function which is given by the k-fold convolution of $g^h(\,)$.

a_k : A random variable indicating the location (time instant t) of the first $\mathcal{H}$-packet arrived over an unexamined interval of length k; the value of a_k is equal to the number of slots between t and the current time.

b_k : Same as a_k applied to $\mathcal{L}$-packets.

$c^l(i,j)$: A random variable describing the cumulative delay of all $\mathcal{L}$-packets which are transmitted between t_k, when $\{r_k\}_{k\ge1}$ is in state (i,j), and the first time in the future when the system becomes empty. In other words, $c^l(i,j)$ denotes the number of $\mathcal{L}$-packets transmitted over $y(i,j)$, $i\ge0$, $j\ge0$, $i+j\ne0$.

$c^l(0,0)$: As before, referring to a period $x=y(0,0)$.

$C^l(i,j)$: Expected value of $c^l(i,j)$, $i\ge0$, $j\ge0$.

$c^h(i,j)$: Same as $c^l(i,j)$ applied to $\mathcal{H}$-packets, $i\ge0$, $j\ge0$.

$C^h(i,j)$: Expected value of $c^h(i,j)$, $i\ge0$, $j\ge0$.

At this point a procedure is developed for the computation of $Y(0,0)=X$. Although the latter quantity may be computed from (2), an alternative computation approach is followed, for two reasons. First, the bounds computed through this approach are required for the derivation of tight bounds on the mean packet delay, as presented in the last section. Second, it is conceptually easier to present this approach by applying it to the process $y(i,j)$, $i\ge0$, $j\ge0$. Based on this approach and some of the derived results, C^l and C^h will then be computed in a straightforward manner. The latter quantities are required for the mean packet delay calculation in (5).

It is easy to establish that the length of the time interval between two consecutive instants (slot boundaries) in which the system is empty is given by (see Fig. 3(a)).

$$y(0,0) = x = \begin{cases} 1 & \text{if } a_1+b_1=0 \\ 1+h_1+y(b_1,h_1) & \text{if } a_1+b_1\ne0 \end{cases} \qquad (6a)$$

where (see Fig. 3(b)) for $i\ge1$, $j\ge0$,

$$y(i,j) = \begin{cases} l_c & \text{if } b_{i-1+j+l_c}+h_{j+l_c}=0 \\ l_c+h_{j+l_c}+y(b_{i-1+j+l_c},h_{j+l_c}) & \text{otherwise} \end{cases} \qquad (6b)$$

and (see Fig. 3(c)) for $i=0$, $j\ge1$,

$$y(i,j) = \begin{cases} 0 & \text{if } a_j+b_j=0 \\ y(1,b_j-1) & \text{if } b_j>a_j\,,\ b_j>0 \\ h_c+y(b_1,a_j-1+h_c) & \text{if } a_j\ge b_j\,,\ a_j>0 \end{cases} \qquad (6c)$$

Equation (6a) is easily explained by considering Fig. 3(a). Let t be a (slot boundary) discrete-time instant at which the buffer is empty. The next such time instant will be t' if no arrivals take place over the slot (t,t'), that is, if $a_1+b_1=0$; in this case $x=1$. If h_1 $\mathcal{H}$-packets arrive over (t,t'), that is if $a_1>0$, then these packets will be transmitted over the next h_1 time slots. The completion time, t'', of these transmissions

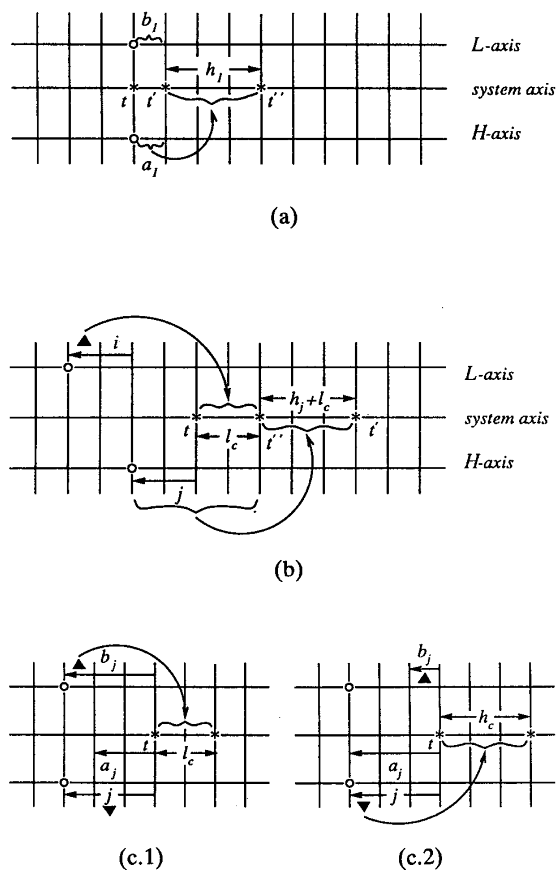

Fig. 3 Illustration of the derivation of the equations in (6); mark * indicates a potential time instant in $\{t_k\}_{k\ge0}$; mark o indicates the boundary between the examined (on the left) and the unexamined (on the right) intervals on the corresponding axes; mark Δ indicates packet arrivals; system axis is the real time axis.

corresponds to a time instant from the sequence $\{t_k\}_{k\geq 0}$, on which the Markov chain $\{r_k\}_{k\geq 1}$ has been defined. At time t'' the unexamined interval on the $\mathcal{H}$-axis is h_1. If a group of $\mathcal{L}$-packets have arrived over the examined interval of the $\mathcal{H}$-axis (t,t'), then $b_1=1$; $b_1=0$ if no such packets have arrived over that interval. Thus, the state of the system can be defined to be (b_1,h_1). By definition, $y(b_1,h_1)$ slots are required for the system to become empty after t''. Thus, if $a_1+b_1\neq 0$, the time required for the system to reach an empty buffer state, starting from an empty buffer state at t, is given by $1+h_1+y(b_1,h_1)$. The values of $y(i,j)$, $i\geq 0$, $j\geq 0$ and $i+j\neq 0$ are given by the linear equations 6(b) and 6(c).

When $i>0$, there is always some $\mathcal{L}$-packet to be served after instant $t\in\{t_k\}_{k\geq 0}$ (Fig. 3). Let $l_c>1$ denote the number of these packets. If no $\mathcal{H}$-packet arrived over the unexamined interval of length $j+l_c$ $(h_{j+l_c}=0)$ and no $\mathcal{L}$-packet arrived over the unexamined interval $i-1+j+l_c$ $(b_{i-1+j+l_c}=0)$ then the system is empty at t'' and, thus, $y(i,j)=l_c$. If, on the other hand, $h_{j+l_c}\neq 0$, then the $\mathcal{H}$-packets which arrived over the interval of length $j+l_c$ are served. The time instant of the completion of this service, t', corresponds to a point in $\{t_k\}_{k\geq 1}$. Thus, in this case, $y(i,j)$ equals $l_c+h_{j+l_c}+y(b_{i-1+j+l_c},h_{j+l_c})$. When, $h_{j+l_c}=0$ but $b_{i-1+j+l_c}\neq 0$ then $t''\in\{t_k\}_{k\geq 1}$ and $y(i,j)$ is equal to $l_c+y(b_{i-1+j+l_c},0)$.

When $i=0$, $j\geq 1$, then no $\mathcal{L}$-packet, which arrived over the examined interval on the $\mathcal{H}$-axis, is in $\mathcal{L}-Q$. Since $t\in\{t_k\}$, the server will serve the oldest group of packets independently of the class in which they belong, according to step (f) of the priority policy. If no packet arrived over the interval of length j, then t corresponds to an empty buffer time instant and thus, $y(0,j)=0$. If a group of $\mathcal{L}$-packets have arrived first over that interval, that is, if $b_j>a_j$, $b_j>0$ (Fig. 3(c.1)), then the server moves to $\mathcal{L}-Q$ at t and the state at that instant can be described as $(1,b_j-1)$. Thus, $y(0,j)=y(1,b_j-1)$. If a group consisting of h_c $\mathcal{H}$-packets have arrived first (or simultaneously with some group of $\mathcal{L}$-packets) over the interval of length j, that is, if $a_j>b_j$, $a_j>0$ (Fig. 3(c.2)), then the server serves the group of the $\mathcal{H}$-packets and then is ready to move to $\mathcal{L}-Q$. Thus, $y(0,j)=h_c+y(b_1,a_j-1+h_c)$, where $b_1=1$ if a group of $\mathcal{L}$-packets arrived simultaneously with that group of $\mathcal{H}$-packets and $b_1=0$ otherwise.

By applying the expectation operator to both sides of the equations in (6), the following infinite dimensional system of linear equations is obtained.

$$Y(i,j)=e(i,j)+\sum_{i_1=0}^{\infty}\sum_{i_2=0}^{\infty} b(i,j,i_1,i_2)Y(i_1,i_2) \qquad (7)$$

The constants $e(i,j)\geq 0$ and $b(i,j,i_1,i_2)\geq 0$ for $i,j,i_1,i_2\geq 0$, may be found in [17]. The following Lemma provides for the existence of a nonnegative and finite solution to the system in (7). Its proof is evident in view of the fact that the solutions of (7) are nonnegative and finite, under stability conditions, [12].

Lemma 3

If $\lambda^l+\lambda^h<1$, then the system in (7) has a unique nonnegative and finite solution. □

The following Theorem provides for a lower bound on $X=Y(0,0)$.

Theorem 2

For $\lambda^h+\lambda^l<1$, a lower bound on $Y=X=Y(0,0)$ is given by $Y_{lo}=Y_{lo}(0,0)$, where $Y_{lo}(0,0)$ is the solution for $Y(0,0)$ of the finite system of linear equations

$$Y(i,j)=e(i,j)+\sum_{i_1=0}^{N_1}\sum_{i_2=0}^{N_2} b(i,j,i_1,i_2)Y(i_1,i_2) \qquad (8)$$

for some $N_1<\infty$ and $N_2<\infty$, where $Y_{lo}(0,0)$ increases monotonically to $Y(0,0)$ as N_1, $N_2\to\infty$.

Proof

For $\lambda^h+\lambda^l<1$, the infinite dimensional system in (7) has a unique nonnegative finite solution (Lemma 3). Thus, the truncated version of (7), shown in (8), has solutions, $Y_{lo}(i,j)$, which satisfy $Y_{lo}(i,j)\leq Y(i,j)$ and $\lim_{N_1,N_2\to\infty} Y_{lo}(i,j)=Y(i,j)$, for $0\leq i\leq N_1$, $0\leq j\leq N_2$, [12].

Theorem 2 establishes a lower bound, $Y_{lo}(0,0)$, on $Y(0,0)$. Numerical results verify that for sufficiently large N_1, N_2, $Y_{lo}(0,0)$ is very close to the true value X, as given by (2).

The previous approach can be applied directly for the calculation of lower bounds on C^h and C^l, which are unknown. These quantities are computed from the following equations, which are easily derived in view of the equations in (6).

$$c^l(0,0)=\begin{cases}0 & \text{if } a_1+b_1=0\\ c^l(b_1,h_1) & \text{if } a_1+b_1\neq 0\end{cases} \qquad (9a)$$

where, for $i\geq 1$, $j\geq 0$,

$$c^l(i,j)=\begin{cases}(i+j-1)l_c+\frac{1}{2}l_c(l_c+1), & \text{if } b_{i-1+j+l_c}+h_{j+l_c}=0\\ (i+j-i)l_c+\frac{1}{2}l_c(l_c+1)+c^l(b_{i-1+j+l_c},h_{j+l_c}), & \text{otherwise}\end{cases} \qquad (9b)$$

(where, $\frac{1}{2}l_c(l_c+1)=1+2\ldots+l_c$) and, for $i=0$, $j\geq 1$,

$$c^l(i,j)=\begin{cases}0 & \text{if } a_j+b_j=0\\ c^l(1,b_j-1) & \text{if } b_j>a_j,\ b_j>0\\ c^l(b_1,a_j-1+h_c) & \text{if } a_j\geq b_j,\ a_j>0\end{cases} \qquad (9c)$$

The corresponding set of equations for $c^h(i,j)$ for $i\geq 0$, $j\geq 0$ are given by

$$c^h(0,0) = \begin{cases} 0 & \text{if } a_1+b_1=0 \\ \frac{1}{2}h_c(h_c+1) + c^h(b_1,h_1) & \text{if } a_1+b_1\neq 0 \end{cases} \quad (10a)$$

where, for $i\geq 1$, $j\geq 0$,

$$c^h(i,j) = \begin{cases} 0 \text{ , if } b_{i-1+j+l_c}+h_{j+l_c}=0 \\ \frac{1}{2}(l_c+j-1)h_{j+l_c}+\frac{1}{2}h_{j+l_c}(h_{j+l_c}+1)+ \\ \quad +c^h(b_{i-1+j+l_c},h_{j+l_c}) \text{ , otherwise} \end{cases} \quad (10b)$$

and, for $i=0$, $j\geq 1$,

$$c^h(i,j) = \begin{cases} 0 \text{ , if } a_j+b_j=0 \\ c^h(1,b_j-1) \text{ , if } b_j>a_j \text{ , } b_j>0 \\ (a_j-1)h_c+\frac{1}{2}h_c(h_c+1)+c^h(b_1,a_j-1+h_c) \\ \quad \text{, if } a_j\geq b_j \text{ , } a_j>0 \end{cases} \quad (10c)$$

The explanation of some of the above equations is given in [17]. By applying the expectation operator to both sides of the equations, two sets of infinite dimensional systems of linear equations are obtained. These systems of equations are of the form of that shown in (7). In fact, the coefficients of the unknowns are identical to those in (7). It is only the constants which are different from e(i,j), i, j,≥0; these constants are denoted by $e^l(i,j)$ or $e^h(i,j)$ and they may be found in [17]

Tight lower bounds on $C^l=C^l(0,0)$ and $C^h=C^h(0,0)$, denoted by C^l_{lo} and C^h_{lo}, respectively, can be obtained by solving truncated versions of the corresponding infinite dimensional systems of linear equations (Theorem 2). Under the stability conditions for the queue, $C^l(i,j)$ and $C^h(i,j)$ is finite since Y(i,j) are finite, for all finite i and j. Then, lower bounds on the mean packet delay can be obtained for each priority class from (see (5))

$$D^h_{lo}=\frac{1-\rho}{\lambda^h}C^h_{lo} \quad \text{and} \quad D^l_{lo}=\frac{1-\rho}{\lambda^l}C^l_{lo} \qquad (11)$$

Finally, upper bounds on the mean packet delay for each class can be obtained from (4), provided that the mean packet delay for the equivalent FIFO queueing system is known. The latter quantity is given by,

$$D^{FIFO}=1+\frac{\sigma-\lambda}{2\lambda(1-\lambda)} , \qquad (12)$$

where σ denotes the second moment of the cumulative number of packet arrivals per slot. Equation (12) is a known result which can be obtained, for instance, by applying the analysis in [16] for independent packet arrival processes.

IV. NUMERICAL RESULTS

In this section the performance of the proposed priority policy is evaluated, in terms of the induced mean packet delay for each priority class. Since the derivation of exact results requires the solution of infinite dimensional systems of linear equations, upper and lower bounds on the induced mean packet delay are computed. To improve the accuracy of the computed results, some techniques are developed for the derivation of tight bounds. The new developments will be presented with respect to quantities associated with the $\mathcal{L}$-packets, but they hold for the corresponding quantities associated with the $\mathcal{H}$-packets, as well. The following definitions are useful for the discussion of this section. Let $\boldsymbol{R}=\{(i_1,i_2)$: $0\leq i_1\leq N_1$, $0\leq i_2\leq N_2\}$ denote the solution region for the system of linear equations in (8); let $\bar{\boldsymbol{R}}$ denotes its complement. Let $\boldsymbol{D}=\{(i_1,i_2)$: $i_1=N_1$ and / or $i_2=N_2\}$ denote the boundary of the solution region $\boldsymbol{R}$.

When the traffic load $\lambda=\lambda^h+\lambda^l$ is small or moderate (e.g., $\lambda<.6$), then the computed lower bounds C^h_{lo} and C^l_{lo} are very close to the true values of $C^h=C^h(0,0)$ and $C^l=C^l(0,0)$ This is justified in view of the observed tightness of the resulting lower and upper bounds on D^h and D^l, computed from (11) and (4). C^h_{lo} and C^l_{lo} are computed by solving (8) for some finite N_1 and N_2, where e(i,j) is replaced by $e^h(i,j)$ and $e^l(i,j)$, $0\leq i\leq N_1$ and $0\leq j\leq N_2$, respectively. The results for $N_1=100$, $N_2=15$ and symmetric packet traffic, defined by $g_c(1)=.5$, $g_c(2)=.25$, $g_c(3)=.25$, are shown in Fig. 4. Notice that the bounds become loose for total packet rate greater than .95. The latter is shown in more detail on Fig. 5 for D^l and Fig. 6 for D^h (curve (4)). The observed (erroneous) decrement of the lower bounds, as the load increases beyond .95, is due to the truncation effect on the infinite dimensional system in (7). This behavior of D^l_{lo} is explained in the

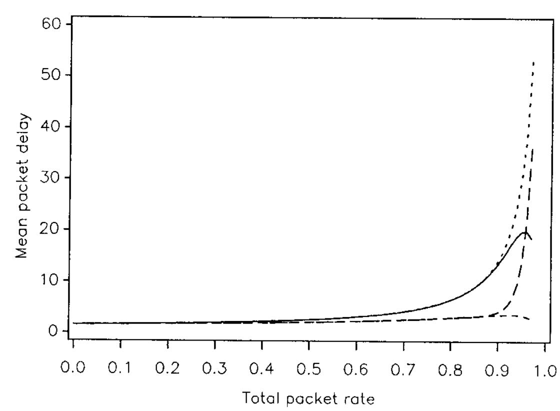

Fig. 4

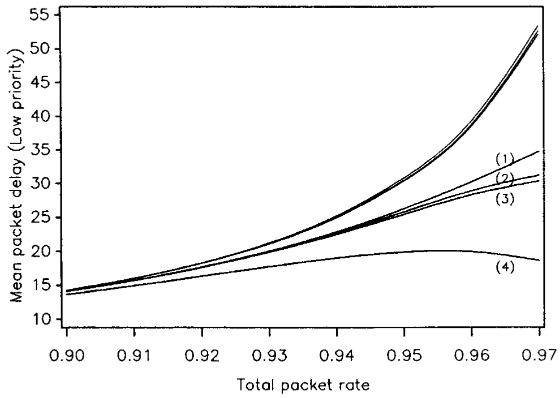

Fig. 5

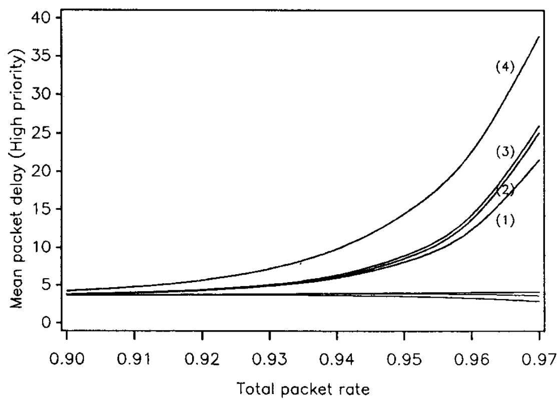

Fig. 6

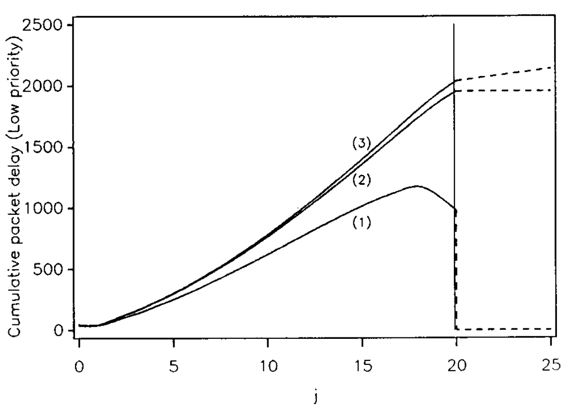

Fig. 7

next paragraph. The behavior of the bounds on D^h can be similarly explained.

From the definition of $C^l(i,j)$ it is evident that the following monotone behavior is expected: for some fixed i_0 and j_0, $C^l(i_0,j_1) \leq C^l(i_0,j_2)$ for $j_1 \leq j_2$ and $C^l(i_1,j_0) \leq C^l(i_2,j_0)$ for $i_1 \leq i_2$. This behavior is clearly not present in the results computed by (8). For instance, a typical behavior of $C^l_{lo}(i_0,j)$ for $0 \leq j \leq N_2$ is shown in Fig. 7 (curve (1)), for $i_0=0$, $N_1=60$ and $N_2=20$. The values of $C^l_{lo}(i_0,j)$ for j close to the truncation boundary $\boldsymbol{D}$ ($j=N_2$) decrease, as j increases. This is easily explained in view of the fact that the values of $C^l(i_0,j)$ for $(i_0,j)\epsilon\boldsymbol{R}$ ($j>N_2$), which affect the computation of the values of $C^l_{lo}(i_0,j)$ close to the boundary $\boldsymbol{D}$, are set equal to zero in (8). When the traffic load is low or moderate and N_1, N_2 are sufficiently large, then the states (i,j) which are close to the boundary $\boldsymbol{D}$ are visited very rarely. As a result, their (inaccurate) contribution to the computed value $C^l_{lo}(0,0)$ is insignificant and the resulting bounds on D^l are tight.

For large traffic loads (e.g., $\lambda>.9$) the boundary effect becomes significant and the bounds become very loose, unless N_1 and N_2 are very large. Under such traffic conditions, the values of $C^l(i,j)$ for $(i,j)\epsilon\bar{\boldsymbol{R}}$ are set to be equal to the value $C^l(\hat{i},\hat{j})$, instead of zero; $(\hat{i},\hat{j})$ denotes the state on $\boldsymbol{D}$ with the minimum distance from (i,j). The latter is implemented by increasing the values of $b(i,j,i_1,i_2)$ for $(i_1,i_2)\epsilon\boldsymbol{D}$, to include the weight of the coefficients $b(i,j,i_1,i_2)$ for $(i_1,i_2)\epsilon\bar{\boldsymbol{R}}$. The resulting new coefficients $\bar{b}(i,j,i_1,i_2)$ may be found in [17]. The values of $\bar{C}^l_{lo}(i,j)$ computed from (8), by incorporating the new coefficients $\bar{b}(i,j,i_1,i_2)$, present the monotone behavior of the exact values $C^l(i,j)$, obtained from the solution of (7). A typical behavior is shown in Fig. 7 (curve (2)). The smaller the solution region $\boldsymbol{R}$ of (8), the larger the improvement on the tightness of the bounds on D^l achieved by utilizing the increased values of the coefficients on the boundary $\boldsymbol{D}$. The values $\bar{C}^l_{lo}(i,j)$ computed from this approach are lower bounds on the true values $C^l(i,j)$, since lower bounds on $C^l(i_1,i_2)$ have been used for $(i_1,i_2)\epsilon\bar{\boldsymbol{R}}$. This is formally proven in [17]. The resulting improvement on the bounds on the delay is shown in Fig. 5 (curve (3)).

Further improvement on $\bar{C}^l_{lo}(i,j)$ can be achieved by boosting the values of $C^l(i,j)$ for $(i,j)\epsilon\bar{\boldsymbol{R}}$. This is achieved by setting the values of $C^l(i,j)$ for $(i,j)\epsilon\bar{\boldsymbol{R}}$ to be equal to the value of $C^l(\hat{i},\hat{j})$, plus a term which increases with i and / or j; this term is a lower bound on the difference between the true value $C^l(i,j)$ and the representing value $C^l(\hat{i},\hat{j})$. This approach is described in [17], where the resulting increased constants $\hat{e}^l(i,j)$, $0 \leq i \leq N_1$, $0 \leq j \leq N_2$, are computed. Again, the values $\hat{C}^l_{lo}(i,j)$ computed from this approach are lower bounds on $C^l(i,j)$, since lower bounds on $C^l(i_1,i_2)$ for $(i_1,i_2)\epsilon\underline{R}$ are used. Curve (3) in Fig. 7 shows the improved values of $C^l(0,j)$ for $0\leq j\leq 20$. Curve (2) in Fig. 5 shows the resulting lower bound on D^l.

Finally, tight bounds on D^l and D^h can be obtained from the expressions

$$D_d^l=\frac{\tilde{C}_{lo}^l}{\lambda^l\tilde{Y}_{lo}} \quad , \quad D_d^h=\frac{\tilde{C}_{lo}^h}{\lambda^h\tilde{Y}_{lo}} \qquad (13)$$

where $\tilde{C}_{lo}^l$, $\tilde{C}_{lo}^h$ and $\tilde{Y}_{lo}$ are computed by using constant (nonzero) values for $C^l(i,j)$, $C^h(i,j)$ and $Y(i,j)$ for $(i,j)\epsilon\underline{R}$, as explained above. By using the concept of dominant systems it is easily shown that D_d satisfies the following inequalities (see [17]).

$$D_{lo}^l\leq D_{dom}^l\leq D^l \quad , \quad D_{lo}^h\leq D_{dom}^h\leq D^h \qquad (14)$$

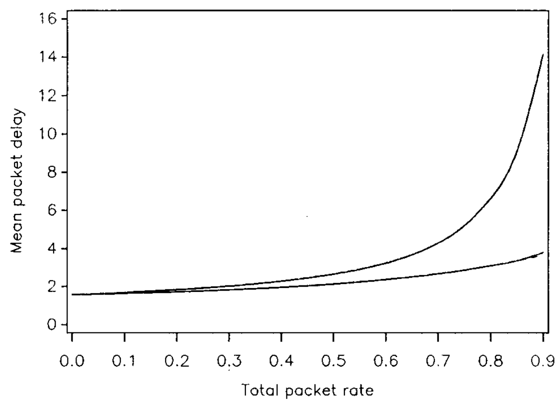

Fig. 8

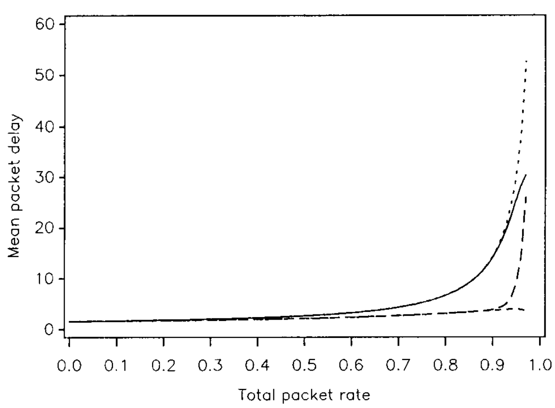

Fig. 9

Curve (1) in Fig. 5 shows the resulting improvement in the lower bound. The bounds on C^l and C^h computed from (14) are the tightest for the case considered here. The mean delay results for each priority class, computed from (14), are shown in Fig. 8, for $0\leq\lambda\leq.90$, and Fig. 9 for $0\leq\lambda\leq 1$. Notice that the upper and lower bounds coincide for $\lambda<.90$, in this case.

REFERENCES

[1] M. Reiser, "Performance Evaluation of Data Networks", Proceedings of the IEEE, Vol. 70, No. 2, February 1982.

[2] H. Kobayashi, A. Konheim, "Queueing Models for Computer Communications System Analysis", IEEE Transactions on Communications, Vol. 25, No. 1, Jan. 1977.

[3] D. Bertsekas, R. Gallager, "Data Networks", Prentice Hall, 1987.

[4] L. Kleinrock, "Queueing Systems", Vol. I & II, John Wiley & Sons, 1976.

[5] R. Wolf, "Stochastic Modeling and the Theory of Queues", Prentice Hall, 1989.

[6] G. Barberis, "A Useful Tool in the Theory of Priority Queueing", IEEE Transactions on Communications, Vol. 28, No. 9, Sep. 1980.

[7] I. Rubin, "Message Delay Analysis of Multiclass Priority TDMA, FDMA, and Discrete-Time Queueing Systems", IEEE Transactions on Information Theory, Vol. 35, No. 3, May 1989.

[8] I. Stavrakakis, "Statistical Multiplexing under Non-i.i.d. Packet Arrival Processes and Different Priority Policies", Performance Evaluation Journal (North Holland), to appear.

[9] C. Bisdikian, " A Queueing Model with Application to Bridges and the DQDB (IEEE 802.6) MAN", Research Report RC 15218, IBM Thomas J. Watson Research Center, Yorktown Heights, Dec. 1989.

[10] R. Newman, Z. Budrikis, J. Hullett, "The QPSX Man", IEEE Communications Magazine, Vol. 26, No. 4, Apr. 1988.

[11] L. Schrage, "An Alternative Proof of a Conservation Law for the Queue G/G/1", Operations Research, 18, pp. 185-187, 1970.

[12] L. Kantorovich, V. Krylov, " Approximate Methods of Higher Analysis", New York, Interscience, 1958.

[13] L. Georgiades, L. Merakos, P. Papantoni-Kazakos, "A Method for the Delay Analysis of Random Access Algorithms whose Delay Process is Regenerative", Journal on Selected Areas in Communications, Vol. 5, No. 6, Jul. 1987.

[14] I. Stavrakakis, D. Kazakos, "A Limited Sensing Protocol for Multi-user Packet Radio Systems", IEEE Transactions on Communications, April 1989.

[15] L. Georgiades, P. Papantoni Kazakos, "Limited Sensing Algorithms for the Packet Broadcast Channel", IEEE Transactions on Information Theory, Vol. 31, pp 280-294, Mar. 1985.

[16] I. Stavrakakis, "Analysis of a Statistical Multiplexer under a General Input Traffic Model", IEEE INFOCOM'90, June 5-7, 1990, San Francisco.

[17] I. Stavrakakis, "Delay Bounds on a Queueing System with Consistent Priorities" IEEE Transactions on Communications, submitted.