J. Austral. Math. Soc. Ser. B 31(1989), 219–239

APPLICATIONS OF MATRIX-GEOMETRIC SOLUTIONS FOR QUEUEING PERFORMANCE EVALUATION OF A HYBRID SWITCHING SYSTEM

MOSHE ZUKERMAN[1]

(Received December 1988; revised February 1989)

Abstract

We consider a hybrid switch which provides integrated packet (asynchronous) and circuit (isochronous) switching. Queue size and delay distribution of the packet switched traffic in the steady state are derived by modelling the packet queue as a queue in a Markovian environment. The arrival process of the packets as well as of the circuit allocation requests are both modelled by a Poisson process. The analysis is performed for several circuit allocation policies, namely repacking, first-fit (involving static or dynamic renumbering) and best-fit. Both exact results and approximations are discussed. Numerical results are presented to demonstrate the effect of increase in packet and circuit loading on the packet delay for each of the policies.

1. Introduction

During the initial phases of the evolving Broadband Integrated Services Digital Networks (B-ISDN) era, based on Fast Packet Switching (FPS) technology, delay sensitive services like video and voice may require special transmission channels (e.g., isochronous) so that a required level of grade of service can be provided.

As a result, hybrid systems which involve an isochronous mode (for delay sensitive services like video and voice) as well as an asynchronous mode (for high speed data communications) are considered. In fact, it is becoming increasingly apparent that such systems will take the lead in providing B-ISDN services before the pure FPS systems will be available. For example, the current status of the IEEE 802.6 [15] evolving standard for Metropolitan

[1]Telecom Australia Research Laboratories, Clayton, Victoria, Australia.

https://doi.org/10.1017/S0334270000006603 Published online by Cambridge University Press

Area Networks (MANs) or Local Area Networks (LANs) involves both modes of transmission.

In this paper, we consider a model of a hybrid communications switch, that provides integrated packet (asynchronous) and circuit (isochronous) switching. Its capacity is dynamically allocated to meet the demand of the synchronous circuits and the remaining capacity is available for the packet switched traffic.

Circuit requests are either accepted or rejected, whereas the packets may be queued. We consider the wastage incurred when practical circuit allocation policies are implemented. Models of hybrid switches in which such waste is not considered have been extensively studied (see for example [17, 18, 27, 31, 32]). In Section 3 we describe several circuit allocation schemes, namely repacking, first-fit (involving static and dynamic renumbering) and best-fit. In [35, 36], we obtain exact results for the statistics of the packet capacity as a function of circuit loading. In this paper, we analyse the packet queue by modelling it as a queueing system in a Markovian environment.

During a heavy traffic period, it is possible (due to traffic burstiness and fluctuations in the capacity made available for packets) that for certain time intervals the service rate is lower than the arrival rate, although the *average* service rate during the entire period is higher than the average arrival rate. In this case, we can say that the queue is *locally unstable* [23], while it may be at the same time *globally stable.* The periods in which the queue is unstable are referred to as *overload* periods.

Avoiding overload periods is especially important in hybrid switching systems. This is due to be the fact that the duration of packet overload caused by the excess capacity used by circuit switching is in the order of minutes, and constitutes a very long time duration relative to the packet service and arrival rates which are in the order of milliseconds. As a result, the packet queue falls under the category of a queueing system with slowly varying service rates [10, 33], where local instability causes low performance. The analysis presented in this paper is used as a tool for design and dimensioning, and for obtaining an optimal flow control policy aimed at avoiding local instability.

If overload modes are avoided, the packet queueing performance under slowly varying service rates can be accurately evaluated by using the *quasi-static* approach [20]. That is, the average performance measure is obtained by a weighted average of measures obtained separately for each capacity mode, where the weights are the steady state probabilities of the process for each mode [18, 20]. Clearly, if overload modes exist (when the queue is locally unstable), such an approach cannot be used.

Under local instability, accurate results for the statistical characteristics of the packet queue cannot be obtained within a reasonable time using simulations. Notice that thousands of packet events are required for each circuit event. Moreover, for a typical sized system with the framing structure discussed in this paper, millions of packet events are required for each change in the packet capacity. Under such conditions, given the state of the art in queueing theory, only matrix-geometric methods [22, 23, 24] may be successfully used; these methods are indeed implemented in Sections 5 and 6.

However, as will be discussed below, due to the 'curse of dimensionality'[2], these methods provide exact results for problems in which the state space is relatively small. When the problem is of a larger size, approximations are required. In Sections 7 and 8 we propose some approximations which unfortunately can be tested only by actual experiment.

2. The framing structure

We consider the hybrid switching system described in [34, 35, 36, 37] which is based on a position multiplexed framing structure as depicted in Figure 1. Such a framing structure represents the framing structure in FDDI-II [4, 19, 13] (a LAN proposed standard with ANSI X3T9.5) as well as in the original proposal of QPSX [1, 2, 13, 25, 26] (with IEEE 802.6). Recently, the 802.6 committee have decided to adopt slot label multiplexing instead of position multiplexing [15]. However, since octets are still position multiplexed within a slot, the model analysed here may still be applicable as an approximation.

Time is divided into intervals of length τ (typically, $\tau = 125$ microseconds), known as frames, each of which is comprised of M equal slots. Within each frame, some of the slots are allocated to synchronous circuits. These slots are designated *isochronous.* Since 8 bit PCM voice samples, arriving once every frame, are usually much smaller than the amount of information transmitted within a time slot are further subdivided into a header plus L octets. In each way, each octet may be used for a 64 kbit voice channel. Hence, each isochronous slot can be used for L different synchronous channels.

The slots which are not used for the isochronous traffic may be used for transmission of packet switched traffic. These slots are designated *asynchronous.* Each of the asynchronous slots is used for transmission of a portion of a packet called a *segment* or a *cell*.

[2]Richard Bellman.

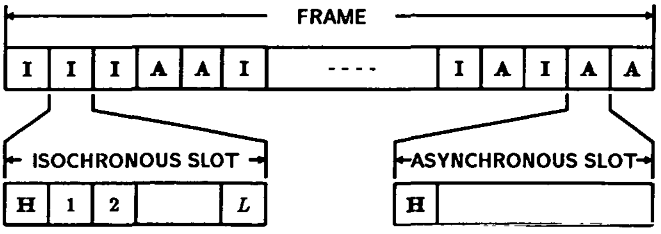

FIGURE 1. A position multiplexed framing structure.

A certain amount of capacity is always reserved for high priority asynchronous traffic, e.g. signalling and control packets. Let M_A be the number of slots reserved for such traffic within a frame. Thus, only $M_I = M - M_A$ slots per frame are available for circuit switching. Therefore, the total number of octets available for the isochronous traffic, denoted by K, is given by $K = LM_I$. The framing structure is illustrated in Figure 1, where **H** stands for a header, **I** for an isochronous slot, and **A** for an asynchronous slot. The L octets within an isochronous slot are represented by $1, 2, \ldots, L$.

Even if only one octet, out of the L available in a slot, is allocated to circuit switching, the entire slot is designated isochronous, thus becoming unavailable for the asynchronous (packet switched) traffic. The waste incurred due to this framing structure may be significant, and demands that efficient circuit allocation and overload control policies are adopted.

Consider all the octets and the slots within a frame (excluding overheads) to be identified by sequentially allocating identification (id) numbers to them in the natural order; i.e., slot number i is the slot that contains the octets with id numbers $(i-1)L+1$ to iL. Let O_m be the maximal id number of all the busy isochronous octets within the mth frame. The wasted octets with id number smaller than O_m are called the *holes* (see [5]). Only part of the waste is due to the holes; the rest is due to *slot-remainder* octets, i.e., the wasted octets with id numbers higher than O_m in the slot containing octet number O_m.

In Figure 2, we present an example with $M = 5$ and $L = 5$ for bandwidth allocation in which a header is again designated by **H**, **b** stands for a busy isochronous octet, **h** for a hole, and **r** for a slot remainder octet.

In this example, we have circuit connections in octets numbered 1, 3, 4, 5, 6, 7, 8, 10, 11 and 12 ($O_m = 12$), holes in octets 2 and 9, and octets 13, 14 and 15 are classified as slot remainder octets.

https://doi.org/10.1017/S0334270000006603 Published online by Cambridge University Press

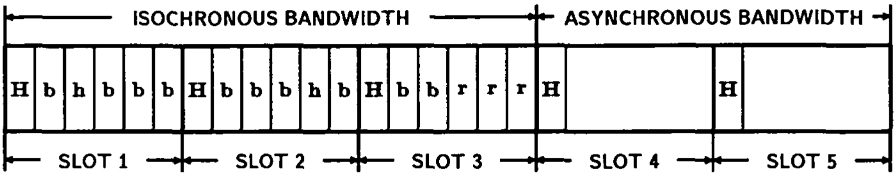

FIGURE 2. An example for bandwidth utilisation.

3. Circuit allocation policies

In this section, we list and describe several circuit allocation policies for which the packet queueing performance is evaluated in this paper.

We start by describing the scheme studied by Fischer and Harris [9] in which only slot-remainder octets were considered. This occurs when no octet is busy while another octet with a smaller id number is free. That is, the number of holes is zero at all times. This can be achieved by reassigning the busy isochronous octets at each termination of a circuit connection. This idealistic circuit allocation policy will be called here *repacking*. Notice that under repacking, the situation described in Figure 2 could not occur. The circuit connection in octets numbered 11 and 12 would be reassigned to octets 2 and 9, and as a result, slot 3 would become asynchronous, thus increasing the packet capacity by 50%. Unfortunately, this circuit allocation policy is usually avoided due to its complexity and its excessive processing cost.

The most common circuit allocation policy, and the simplest to implement, is *First-Fit* (FF) [16]. Under this policy, we assign for each incoming circuit request the octet with the smallest id number among all the empty octets. Octets are not reassigned following a termination of a circuit usage, and therefore holes are created and the waste is increased. In [35, 36, 37] we have considered two possible implementations of FF: *FF with Static numbering (FFS)*, and *FF with Dynamic renumbering (FFD)*.

Under FFS, the id numbers assigned to slots stay fixed during the entire operation. The *movable boundary* [17, 31], that is, the boundary between the isochronous and the asynchronous bandwidth, is always located immediately after the slot containing O_m (assuming that the mth frame is under consideration). In this case, it is theoretically possible that an isochronous slot comprised of L consecutive holes is created. This event will henceforth be the *empty slot* event. In practice, this is a very rare event in view of the fact that the IEEE 802.6 has considered L to be large relative to M. ($L = 64$ and M depends on the networks transmission rate, which based on current

https://doi.org/10.1017/S0334270000006603 Published online by Cambridge University Press

technology is limited to 28 plus overheads.) However, for future implementations with higher M, or for implementations with a relatively small L, or under extremely bursty isochronous traffic conditions, this event may occur.

It should be mentioned that the id numbers assigned to the different slots do not have to be based on their position within the frame. In fact, in order to reduce the variance of the packet delay, it is beneficial not to have the isochronous bandwidth continuous in time as in Figure 2. By spreading the isochronous slots all over the frame, we can avoid long packet service interruptions, and improve the packet queueing performance. For example, for the case of $M = 5$ as in Figure 2, we can assign the first slot for the first 5 incoming circuit requests. Then, when slot 1 is completely filled with circuit connections, we assign new incoming circuit requests to slot 3. When both slots 1 and 3 are filled with circuit connections, we assign new ones to slot 5, and finally to slot 2. (Slot 4 is always reserved for the asynchronous traffic.) This order of circuit allocation also falls under the category of FFS, as we assign the numbers 1, 2, 3, 4, 5 to the slots positioned in places 1, 3, 5, 2, 4 within the frame. In fact, as long as the order assigned to isochronous slots is consistent for all frames, such a circuit allocation mechanism falls under FFS.

The empty slot event is avoided utilising FFD. Here again, the assignment of octets to incoming circuit requests is based on FF. However, unlike FFS, in this scheme, as soon as the empty slot event occurs, the relevant slots and octets are renumbered to preclude this situation. In other words, the empty isochronous slot becomes asynchronous, and it is logically considered to be moved to the other side of the movable boundary. In this case, newly arriving requests are assigned first to the empty octets within the isochronous slots so that this slot continues to be asynchronous.

Another circuit allocation scheme, besides FFD, in which the empty slot event is avoided is a scheme called *Best-Fit (BF)*. Under this method (mentioned in [34]), for each incoming circuit request, we assign the octet with the smallest identification number among all the empty octets within the isochronous slot with the most busy octets. As soon as the last circuit request is terminated within an isochronous slot, the slot becomes asynchronous. Under this scheme, the bandwidth is utilised more efficiently than under FF (FFS or FFD), and therefore it should be considered for implementation especially under certain extremely bursty isochronous traffic conditions, or in cases in which L to small relative to M.

In this paper, we obtain exact results for the statistics of the packet queue under repacking, FFS, FFD, and BF. As will be explained later, exact results for FFS, FFD and BF can only be obtained for small sized problems, since the complexity of the solution grows exponentially, unlike the case of repacking in which the growth is linear.

Notice that when the probability of the empty slot event is very small, as is the case in practical applications with $L \geq 30$ [35], the results for FFS, FFD and BF are similar. In such cases, the results for FFS could serve as a very good approximation for FFD and BF. In cases where the probability of the empty slot event is more significant, the results for FFS provide us with an upper bound for certain performance measures, e.g., the average delay or 99th percentile of the delay under FFD. Notice that the results for repacking can always serve as a lower bound for such performance measures. Thus both bounds are available.

In [35], it is demonstrated that the wastage in packet capacity under repacking, due to slot remainder waste, for a system with $M = 8$, $M_A = 1$ and $L = 31$, is within 6–12%. It is also demonstrated that an additional waste of 4–14% is incurred under FF or BF due to holes. (Note that based on [15], L in QPSX is increased to 64, in which case both types of wastage are significantly increased.)

4. The model

The packet queue is modelled as a single server queue in a Markovian environment with unlimited buffer size and fluctuating service rate based on circuit loading.

It is assumed that packet interarrival times, circuit allocation request interarrival times and circuit holding times are independent and exponentially distributed with parameters λ_p, λ_c, and μ_c respectively.

Based on the description of the framing structure in Section 2, we have the number of isochronous slots within the mth frame, denoted by S_m, is a function of two components:

1. $N_m \triangleq$ the number of *busy isochronous octets*, i.e., the number of octets allocated and actually used within the mth frame by the circuit switched traffic.

2. $W_m \triangleq$ the number of wasted octets within the mth frame, including both holes and slot remainder octets.

As a result, $S_m = (N_m + W_m)/L$.

We now define, as in [35], the continuous time processes:

$$S_t \triangleq S_m \qquad \text{for } (m-1)\tau \leq t < m\tau, \qquad m = 1, 2, 3, \ldots,$$
$$N_t \triangleq N_m \qquad \text{for } (m-1)\tau \leq t < m\tau, \qquad m = 1, 2, 3, \ldots,$$

https://doi.org/10.1017/S0334270000006603 Published online by Cambridge University Press

and their limits:

$$S_\infty = \lim_{t\to\infty} S_t = \lim_{m\to\infty} S_m$$
$$N_\infty = \lim_{t\to\infty} N_t = \lim_{m\to\infty} N_m.$$

Since the frame time, namely τ, which is in the order of micro-seconds, is very small in comparison with the circuit holding and interarrival times, the processes S_t, $t \geq 0$ and N_t, $t \geq 0$ can be modelled as continuous time Markov chains. The packet queue is therefore modelled as a queue in a Markovian environment [24].

The packet service time is a function of two elements: the size of the packet, and the state of the Markovian environment (namely S_t) during the service of the packet. We assume that packet service time is exponentially distributed with fluctuating rate based on circuit loading. That is, under $S_t = i$, the infinitesimal packet service rate is given by

$$\mu_p(i) = 64{,}000 \times L \times (M - i)/p_e \text{ packets/sec.,} \tag{1}$$

where p_e represents the average effective packet size in bits.

Before we present the analysis, we find it appropriate to make some comments regarding the simplifying assumptions of exponential service and interarrival times. It has been established that in many actual packet switching systems, the arrival process has the following characteristics: (1) it is of a bursty nature, namely the variance of the interarrival times is high, and (2) the interarrival times are correlated. It has also been established that the exponential distribution does not always represent packet service time. Actually, there exists evidence that in some cases this service time is actually bi-modal in nature [8, 30]. therefore, ideally, a more appropriate model for the arrival process is the Switched Poisson Process (SPP) [11, 12, 21, 29, 38], and for the service time a probability distribution of phase type [24]. The method described in this paper can, in theory, be easily extended to such a model, by considerably increasing the dimension of the Markovian environment. Unfortunately, however, for the problem under consideration, as will be discussed below, even under the simplifying assumptions of exponential service and interarrival times, due to the 'curse of dimensionality', exact queueing results are limited to small problems (e.g. $K \leq 20$), and approximations are required for larger problems.

Therefore, we are faced with the problem of 'approximate now or approximate later' (even for small problems), and in this paper, we choose to approximate now in order to keep the formulation simple, and to be able to present exact results for small problems. Nevertheless, it will be the subject of future research to test the different options of approximations of a more complex model by actual experiment.

5. Repacking

Under repacking, for any frame m, the value of S_m is directly obtained as a function of N_m, by $S_m = \lceil N_m/L \rceil$, where $\lceil x \rceil$ represents the smallest integer greater or equal to x. Therefore, the state of the Markovian environment (as seen by the asynchronous traffic) under repacking can be described by the state of the process $\{N_t, t \geq 0\}$.

The infinitesimal generator of the Markovian environment, for the case of repacking, denoted by the $(K+1) \times (K+1)$ matrix Q, is simply the generator for an Erlang system, and its elements are given as follows:

$$
\begin{aligned}
&Q(0,0) = -\lambda_c \\
&Q(i,i) = -(i\mu_c + \lambda_c) \quad \text{for } i = 1,2,\ldots,K-1 \\
&Q(i,i+1) = \lambda_c \quad \text{for } i = 0,1,\ldots,K-1 \\
&Q(i,i-1) = i\mu_c \quad \text{for } i = 1,2,\ldots,K \\
&Q(K,K) = -K\mu_c \\
&Q(i,j) = 0 \quad \text{otherwise}
\end{aligned}
$$

where $Q(i,j)$ is the i,j element in the matrix Q.

The steady state probability of the Markovian environment being in state i is given by Erlang's first formula,

$$\text{Prob}(N_\infty = i) = A^i / \left\{ i! \sum_{n=0}^{K} [A^n/n!] \right\} \quad \text{for } i = 0,1,2,\ldots,k. \tag{2}$$

where A is the total offered circuit switched traffic in Erlangs, that is, $A = \lambda_c/\mu_c$.

Let Π be the invariant vector of the Markovian environment, that is, Π is a $(K+1)$-vector in which the ith element represents the distribution $\text{Prob}(N_\infty = i)$ for $i = 0,1,2,\ldots,k$.

This model leads to a Quasi Birth and Death process (QBD) [24] with the state space $F = \{(i,j): i \geq 0, 0 \leq j \leq M_I\}$. The index i is the number of packets in the system and j represents the state of the Markovian environment, that is, the state of the process $\{N_t, t \geq 0\}$. The following is based on Neuts [24], and describes how the statistics of the packet queue can be obtained using matrix geometric solutions.

Denote by λ a $(K+1)$-vector all of whose elements have the value λ_p. Let γ be a $(K+1)$-vector representing the service rates γ_i $0 \leq i \leq K$. That is,

γ_i is the infinitesimal packet service rate given that $N_t = i$, and is given by $\gamma_i = \mu_p(\lceil i/L \rceil)$.

Also, denote by $\Delta(\lambda)$, $\Delta(\gamma)$ and $\Delta(\gamma+\lambda)$ the three $(K+1)\times(K+1)$ diagonal matrices in which the diagonals are λ, γ and $\gamma + \lambda$ respectively.

The generator $\tilde{Q}$ of the QBD under consideration, which represents the packet queue, is then given as (6.2.1) in page 258 in [24].

Note that when the finite buffer case is under consideration, that is, when the queue size is of limited size, then the matrix $\tilde{Q}$ is finite, and if its dimension is not too large, the statistics of the asynchronous queue size can be obtained using successive overrelaxation [6].

Base on (6.2.2) in [24], the stability condition of the packet queue is given by

$$\lambda_p < \Pi\gamma^{\mathrm{T}}. \tag{3}$$

Let x_{ij} be the steady state probability of having i packets in the queue, and the Markovian environment in state j $(j = 0, 1, \ldots, K)$. Also, define for each $i \geq 0$ the vector X_i, of dimension $K+1$, as $X_i \triangleq [x_{i0}, x_{i1}, \ldots, x_{iK}]$.

Given that the queue is stable, the stationary probability vector $X = [X_0, X_1, \ldots]$ is given by (6.2.5) in [24] as

$$X_i = \Pi(I - R)R^i, \quad \text{for } i \geq 0 \tag{4}$$

where R is a $(K+1)\times(K+1)$ matrix obtained by a sequence of successive substitutions (starting with $R = 0$) of the following matrix quadratic equation [24].

$$R = [R^2\Delta(\gamma) + \Delta(\lambda)][\Delta(\gamma+\lambda) - Q]^{-1}. \tag{5}$$

Then the steady state probability of having i packets in the packet queue, denoted by P_i, is simply the sum of the elements in the vector X_i, i.e.,

$$P_i = \sum_{j=1}^{M} x_{ij}. \tag{6}$$

Let $\overline{W}(x)$ be the probability that a packet will wait longer than x seconds. The value of $\overline{W}(x)$, obtained by Ramaswami and Lucantoni in [28], is given by

$$\overline{W}(x) = \sum_{n=0}^{\infty} d_n e^{-\theta x}(\theta x)^n/n! \tag{7}$$

where: $\theta = \max_{0\leq j\leq K}\{-B(j,j)\}$, where $B(j,j)$ is the (j,j)th element in the matrix B which is given by $B = Q - \Delta(\gamma+\lambda) + \Delta(\lambda)$; and $d_n = X_0(I-R)^{-1}RH_n\mathbf{e}$, where the matrix H_n is obtained recursively by $H_0 = I$, and $H_{n+1} = H_nU_1 + RH_nU_2$ for $n \geq 0$, where $U_1 = B/\theta + I$ and $U_2 = \Delta(\gamma)/\theta$.

Now we present some numerical results obtained using the above described method for the case of repacking, and compare them with approximations.

https://doi.org/10.1017/S0334270000006603 Published online by Cambridge University Press

We consider a small example with $p_e = 4000.0$, $\mu_c = 1.0/180$ sec.$^{-1}$ (namely circuit holding time of 3 minutes), $M = 4$, $L = 3$, $M_a = 1$; hence, $M_I = 3$ and $k = 9$. The packet service rate provided by a single slot is given by $64,000 \times L/p_e$, and for this example is 48 packets/sec. Since $m_I = 1$, namely one slot is always reserved for the packets, the minimal packet service rate is 48 packets/sec. Since $M = 4$, the maximal packet service rate is $4 \times 48 = 192$ packets/sec. The average packet service rate (ASR) under repacking for this example is given by ASR $= \sum_{i=0}^{k} \gamma_i P(N_t = i) = 105.84$ packets/sec.

In Table 1, we present results for the average delay in seconds as a function of the packet arrival rate where the circuit request arrival rate stays fixed at $\lambda_c = 1.5/60$ arrivals/sec. (namely 1.5 per minute). We also present results for two commonly used approximations. The first is the M/M/1 approximation, in which the delay is computed assuming that the average service rate stays constant during the entire operation and the average delay is computed by the classical formula for the delay under M/M/1, namely,

$$E_{mm1}[D] = (1/\text{ASR})/(1 - \lambda_p/\text{ASR}). \tag{8}$$

The second is the above-mentioned quasi-static approximation, which can be used only when local instability does not occur, namely $\gamma_i > \lambda_p$, $\forall i$, $i = 0, 1, 2, \ldots, K$, in which the average delay is computed by

$$E_{qst}[D] = \sum_{i=0}^{K} \frac{1/\gamma_i}{1 - \lambda_p/\gamma_i} P(N_t = i) \quad \text{under the condition:} \min_i\{\gamma_i\} > \lambda_p. \tag{9}$$

TABLE 1. Exact results, the M/M/1 and the quasi-static approximations, for the average packet delay, under repacking with $\lambda_c = 1.5/60$ sec.$^{-1}$ for different packet arrival rates.

λ_p	$E[D]$	$E_{mm1}[D]$	$E_{qst}[D]$
10.0	0.01242	0.01043	0.01242
20.0	0.01484	0.01165	0.01484
30.0	0.01913	0.01318	0.01914
40.0	0.03127	0.01519	0.03151
50.0	0.53268	0.01791	local instability
60.0	3.56244	0.02181	local instability

The results in Table 1 demonstrate that the quasi-static approximation provides accurate results for the cases in which local instability does not occur. (Note that in our example, local instability occurs for $\lambda_p \geq 48.0$.) It is also demonstrated that the M/M/1 approximation, which unfortunately, is commonly used in practice, is not accurate, especially under local instability. (It is only accurate in cases where an accurate evaluation is not needed, namely, under low utilisation.)

In Table 2, we fix the packet arrival rate at $\lambda_p = 60.0$ packets/sec., and present results for the average packet service rate and delay for different values of the circuit arrival rate. We notice extremely high average delays under local instability although the system is globally stable.

TABLE 2. The Average Service Rate (ASR) and the average delay, $E[D]$, for the asynchronous traffic, under repacking with $\lambda_p = 60.0$ for different circuit arrival rates.

λ_c	ASR	$E[D]$
0.5/60	151.51	0.02162
1.0/60	127.94	0.52203
1.5/60	105.84	3.56243
2.0/60	87.82	11.65052
2.5/60	75.07	28.47007

6. First-fit and best-fit

For both FF and BF the state of the Markovian environment can be described by an M_I dimensional vector $\mathbf{m} = [n_1, n_2, \ldots, n_{M_I}]$, where n_j is the number of busy isochronous octets within slot j.

The total number of states in the Markovian environment, denoted by $\hat{K}$ and given by $\hat{K} = (L+1)^{M_I}$, grows exponentially with m_I. Such a number of states is very large even for a small practical application (e.g., $M = 4$, $M_I = 3$ and $L = 64$). In the next section we shall present an approximation in which the number of states of the Markovian environment grows polynomially with M_I.

For each state in the Markovian environment we define the following:

Let G be the set of all indices $j \in \{1, \ldots, M_I\}$ such that $n_j < L$. That is, G represents the set of slots which are not filled with circuit connections. Also, let g be the minimal index in G.

Let U be the set of all indices $j \in \{1, \ldots, M_I\}$ such that $n_j > 0$. That is, U represents the set of isochronous slots under FFD or BF. In addition, let u and $\hat{u}$ be the minimal and the maximal index in U respectively.

https://doi.org/10.1017/S0334270000006603 Published online by Cambridge University Press

Define $V = U \cap G$ and let v be the minimal index in V.

Let H be the set of all indices $j \in G$ such that $n_j \geq n_i$, $\forall i \in G$, and let h be the minimal element in H.

Then, for FFS, the capacity in terms of number of slots available for the asynchronous traffic is given by $Y(\mathbf{m}) = M - \hat{u}$. For FFD and BF, this capacity is given by $Y(\mathbf{m}) = M - |U|$, where $|X|$ represents the number of elements in the set X.

According to the description of FFS, FFD and BF, we have that a new circuit request arrival will be assigned to slot numbers g, v and h for FFS, FFD and BF respectively.

The infinitesimal generator of the Markovian environment, the matrix Q for the case of FFS, is described in Table 3. (The set G and its minimal index g in Table 3 are with respect to $\mathbf{a}$.)

TABLE 3. The infinitesimal generator of the Markovian environment of FFS.

$\mathbf{a}$	$\mathbf{b}$	$Q(\mathbf{a}, \mathbf{b})$
$[n_1, \ldots n_g, \ldots n_{M_I}]$	$[n_1, \ldots n_g + 1, \ldots n_{M_I}]$	λ_c
$[n_1, \ldots n_j, \ldots n_{M_I}]$	$[n_1, \ldots n_j - 1, \ldots n_{M_I}]$	$n_j \mu_c,\ 1 \leq j \leq M_I$
$[n_1, \ldots, n_{M_I}]$	$[n_1, \ldots, n_{M_I}]$	$-(\lambda_c + \mu_c \sum_{j=1}^{M_I} n_j)$ for $\lvert G\rvert > 0$
$[n_1, \ldots, n_{M_I}]$	$[n_1, \ldots, n_{M_I}]$	$-\mu_c \sum_{j=1}^{M_I} n_j$ for $\lvert G\rvert = 0$
otherwise		0

Similarly, the infinitesimal generator of the Markovian environment, the matrix Q for the case of FFD, is described in Table 4. (The set G and the minimal index v of V in Table 4 are with respect to $\mathbf{a}$.)

TABLE 4. The infinitesimal generator of the Markovian environment of FFD.

$\mathbf{a}$	$\mathbf{b}$	$Q(\mathbf{a}, \mathbf{b})$
$[n_1, \ldots n_v, \ldots n_{M_I}]$	$[n_1, \ldots n_v + 1, \ldots n_{M_I}]$	λ_c
$[n_1, \ldots n_j, \ldots n_{M_I}]$	$[n_1, \ldots n_j - 1, \ldots n_{M_I}]$	$n_j \mu_c, 1 \leq j \leq M_I$
$[n_1, \ldots, n_{M_I}]$	$[n_1, \ldots, n_{M_I}]$	$-(\lambda_c + \mu_c \sum_{j=1}^{M_I} n_j)$ for $\lvert G\rvert > 0$
$[n_1, \ldots, n_{M_I}]$	$[n_1, \ldots, n_{M_I}]$	$-\mu_c \sum_{j=1}^{M_I} n_j$ for $\lvert G\rvert = 0$
otherwise		0

In a similar fashion, the infinitesimal generator of the Markovian environment for the case of BF is described in Table 5. (The set G and the minimal index h in Table 5 are with respect to **a**.)

TABLE 5. The infinitesimal generator of the Markovian environment of BF.

a	**b**	$Q(\mathbf{a}, \mathbf{b})$
$[n_1, \dots n_h, \dots n_{M_l}]$	$[n_1, \dots n_h + 1, \dots n_{M_l}]$	λ_c
$[n_1, \dots n_j, \dots n_{M_l}]$	$[n_1, \dots n_j - 1, \dots n_{M_l}]$	$n_j \mu_c, 1 \le j \le M_l$
$[n_1, \dots, n_{M_l}]$	$[n_1, \dots, n_{M_l}]$	$-(\lambda_c + \mu_c \sum_{j=1}^{M_l} n_j)$ for $\|G\| > 0$
$[n_1, \dots, n_{M_l}]$	$[n_1, \dots, n_{M_l}]$	$-\mu_c \sum_{j=1}^{M_l} n_j$ for $\|G\| = 0$
otherwise		0

It is convenient to write the states of the Markovian environment as a one dimensional vector. Accordingly, we sequentially assign the numbers 0 through $\hat{K}$ to the states of the Markovian environment as follows. State 0 is $[0, 0, 0, \dots, 0, 0]$, state 1 is $[0, 0, 0, \dots, 0, 1]$, state 2 is $[0, 0, 0, \dots, 0, 2]$ and so on, until state $L([0, 0, 0, \dots, 0, L])$. Then state $L + 1$ corresponds to $[0, 0, 0, \dots, 1, 0]$, state $L + 2$ to $[0, 0, 0, \dots, 1, 1]$, etc. The matrix Q is written such that the above order of the states is kept.

Let γ_i be the packet service rate, given that the environment is in state number i. Let $\mathbf{m}(i) = [n_1(i), n_2(i), \dots, n_{M_l}(i)]$, be the vector represented by state number i. Then, γ_i is given by

$$\gamma_i = 64,000 \times L \times Y(\mathbf{m}(i))/p_e \text{ packets/sec.} \tag{10}$$

As in the previous section, we denote by λ a $\hat{K}$-vector all of whose elements have the value λ_p. Let γ be a $\hat{K}$-vector representing the service rates γ_i, $0 \le i \le K$. Accordingly, denote by $\Delta(\lambda)$, $\Delta(\gamma)$ and $\Delta(\gamma + \lambda)$ the three $\hat{K} \times \hat{K}$ diagonal matrices in which the diagonals are λ, γ and $\gamma + \lambda$ respectively. Then, the statistics of the delay are obtained as described in the previous section.

Now we consider again the example presented in the previous section. In Table 6, we present numerical results for the average packet delay under FFS, FFD and BF for different values of λ_p, where the circuit arrival rate stays fixed at $\lambda_c = 1.5/60$ sec.$^{-1}$.

In Table 7, we present results, for the same example, for the average packet service rate and for the average packet delay under FFS, FFD and BF, for different values of λ_c, where the packet arrival rate stays fixed at $\lambda_p = 60.0$ packets/sec.

TABLE 6. The average packet delay, under FFS, FFD and BF with $\lambda_c = 1.5/60$ sec.$^{-1}$ for different packet arrival rates.

λ_p	FFS	FFD	BF
10.0	0.01651	0.01534	0.01525
20.0	0.02088	0.01913	0.01900
30.0	0.02948	0.02643	0.02619
40.0	0.05726	0.04946	0.04886
50.0	3.44625	2.11022	2.01517
60.0	24.24321	13.90689	13.12824

TABLE 7. The Average Service Rate (ASR) and the average delay, $E[D]$, for the asynchronous traffic, under FFS, FFD and BF, with $\lambda_p = 60.0$ packets/sec. for different circuit arrival rates.

λ_c	FFS		FFD		BF	
	ASR	$E[D]$	ASR	$E[D]$	ASR	$E[D]$
0.5/60.0	146.95	0.20873	148.28	0.09461	148.32	0.09269
1.0/60.0	113.74	5.54747	117.51	2.78271	117.73	2.66579
1.5/60.0	86.37	24.24321	91.34	13.90689	91.75	13.12824
2.0/60.0	68.33	56.28569	72.86	34.60860	73.34	32.33195

7. An approximation for FFS

The state of the Markovian environment in the case of FFS can be described by a three dimensional vector (i, j, n), where n represents the number of (possibly empty) isochronous slots, and i and j represent the number of busy isochronous octets within the first $n-1$ slots and within the nth slot respectively.

Therefore, the total number of states of the Markovian environment is given by

$$(L+1)+L\times(L+1)+L\times(2L+1)+L\times(3L+1)+\cdots+L\times[(M_I-1)L+1] = 1+\overline{K},$$

where $\overline{K} = K + \frac{1}{2}M_I(M_I-1)L^2$.

Notice that the number of states of the Markovian environment, although it may be too large for large problems, exhibits polynomial growth.

The infinitesimal generator of the Markovian environment, namely the matrix Q, is irreducible, and its elements are given as in Table 8.

TABLE 8. The infinitesimal generator of the Markovian environment.

$\mathbf{a}$	$\mathbf{b}$	$Q(\mathbf{a},\mathbf{b})$	for
$[i,u,n]$	$[i+1,j,n]$	λ_c	$0 \le i \le L(n-1)-1$, $0 \le j \le L$ and $2 \le n \le M_I$
$[L(n-1),j,n]$	$[L(n-1),j+1,n]$	λ_c	$0 \le j \le L-1$ and $1 \le n \le M_I$
$[i,j,n]$	$[i-1,j,n]$	$i\mu_c$	$1 \le i \le L(n-1)$, $0 \le j \le L$ and $1 \le n \le M_I$
$[i,j,n]$	$[i,j-1,n]$	$j\mu_c$	$0 \le i \le L(n-1)$, $2 \le j \le L$ and $1 \le n \le M_I$
$[L(n-1),L,n]$	$[Ln,1,n+1]$	λ_c	$1 \le n \le M_I - 1$
$[0,0,0]$	$[0,1,1]$	λ_c	
$[i,1,n]$	$[i-j,j,n-1]$	$P(i,j,n-1)\mu_c$	$0 \le i \le L(n-1)$, $0 \le j \le \min\{i,L\}$ and $1 \le n \le M_I$
$[i,j,n]$	$[i,j,n]$	$-\sum_{z\in Z_\mathbf{a}} Q(\mathbf{a},\mathbf{z})$ where $Z_\mathbf{a}$ is the set of all $\mathbf{c}$ such that $Q(\mathbf{a},\mathbf{c})$ is defined above.	$0 \le i \le L(N-1)$, $0 \le j \le L$ and $1 \le n \le M_I$
and zero otherwise			

Here $P(i,j,n-1)$ should be the conditional probability of having j busy isochronous octets in slot $n-1$ given that (1) there are i busy isochronous slots within slots 1 through $n-1$, and (2) the number of busy isochronous slots has just been reduced from n to $n-1$. In order to avoid further increases in the state space we shall eliminate the second conditon and we shall approximate $P(i,j,n-1)$ by the steady state conditional probability of having j busy isochronous octets in slot $n-1$ given that there are i busy isochronous slots within slots 1 through $n-1$.

To compute the statistics of the delay, we again apply the matrix-geometric solutions as described in Section 5. Accordingly, we sequentially assign the numbers 0 through $\overline{K}$ to the states of the Markovian environment. In particular, the states are ordered lexicographically (sorted first by n, then by i, and then by j), i.e., state 0 is $[0,0,0]$, states 1 to L are $[0,1,1]$ to $[0,L,1]$. Then, states $L+1$ to $2L$ are $[0,1,2]$ to $[0,L,2]$, and states $2L+1$ to $(L+2)L$ are $[1,1,2]$ to $[L,L,2]$, and so on until state $\overline{K}$ is $[(M_I-1)L,L,M_I]$.

This way we have a one dimensional state space and the matrix Q is written such that the above order of the states is kept.

We denote by γ_m the service rate of the packets given that the environment is in state number m. Then γ_m is given by

$$\begin{aligned}
\gamma_0 &= \mu_p(0) \\
\gamma_m &= \mu_p(1) \quad \text{for } 1 \le m \le L \\
\gamma_m &= \mu_p(n) \quad \text{for } (1/2)(n-1)(n-2)L^2 + (n-1)L + 1 \\
&\qquad\qquad \le m \le (1/2)n(n-1)L^2 + nL \text{ and } 2 \le n \le M_I.
\end{aligned}$$

As in the previous sections, we denote by λ a $(\overline{K}+1)$-vector all of whose elements have the value λ_p. Let γ denote a $(\overline{K}+1)$-vector representing the service rates γ_i, $0 \le i \le \overline{K}$.

Also, denote by $\Delta(\lambda), \Delta(\gamma)$ and $\Delta(\gamma+\lambda)$ the three $\overline{K} \times \overline{K}$ diagonal matrices in which the diagonals are λ, γ and $\gamma + \lambda$ respectively.

Then the statistics of the delay are obtained as described in Section 5.

8. Phase type fitting

Even utilising the approximation described in the previous section, it is not practical to obtain numerical results for a large network (e.g., $M = 28$ and $L = 64$). In this section, we discuss further approximations which can be utilised to obtain numerical results. The main idea behind these approximations can be described as follows. Notice that the time that the process $\{S_t, t \ge 0\}$ spends in each state, namely *service mode duration*, is phase type distributed [24]. Clearly, the number of phases of this distribution is very large, causing the matrix R to be too large, which makes it impossible to use a brute force application of matrix-geometric solutions. Therefore we shall consider replacing the real phase type distribution of the service mode duration by an approximate one with fewer phases, preserving some of its most important statistics. Then the implementation of matrix-geometric methods is possible and numerical results are obtainable.

Denote by Λ_{D_i} and Λ_{U_i}, for $i = 1, 2, \ldots, M_I - 1$, the time the process $\{S_t, t \ge 0\}$ spends continuously in state i, given that the process $\{S_t, t \ge 0\}$ enters state i from state $i-1$ and $i+1$ respectively. In addition, we define by Λ_{M_I} the time the process $\{S_t, t \ge 0\}$ spends continuously in state M_I. (Clearly, the process $\{S_t, t \ge 0\}$ can enter state M_I only from state $M_I - 1$.)

Notice that, Λ_{D_i} and $\Lambda_{U_i} (i = 1, 2, \ldots, M_I - 1)$, for both repacking and FF, are the time until absorption for a birth and death process with a finite number of states out of which the first and the last state are two absorbing barriers, therefore both Λ_{D_i} and Λ_{U_i} are phase type distributed, with the representation (α_{D_i}, T_{D_i}) and (α_{U_i}, T_{U_i}) respectively, for which the distribution and the moments are given in pages 45 and 46 of Neuts [24].

For the case of repacking, we consider, for each mode $i (i = 1, 2, \ldots, M_I - 1)$ of the process $\{S_t, t \geq 0\}$, the Markov process $\{N_t, t \geq 0\}$ between the barriers $L(i-1)$ and $Li+1$. In this case Λ_{D_i} and Λ_{U_i} are the absorption times starting from states $L(i-1)+1$ and Li respectively. Clearly, Λ_{M_I} is phase type distributed and represents the time until absorption to a single barrier, namely $L(M_I - 1)$, starting from the state $L(M_I - 1) + 1$.

In the case of FFS, based on the approximation presented in Section 7, we consider, for each mode $i (i = 1, 2, \ldots, M_I - 1)$ of the process $\{S_t, t \geq 0\}$, a two dimensional continuous time Markov chain denoted by

$$\{N_1^i, N_2^i\} = \{N_1^i(t), N_2^i(t), t \geq 0\} \quad \text{for } i = 1, 2, \ldots, M_I,$$

where $N_1^i(t)$ and $N_2^i(t)$ denote the number of busy isochronous octets within the first $i-1$ slots, and within octets $L(i-1)+1$ through $Li+1$ at time t, respectively.

The lower barrier for both Λ_{D_i} and $\Lambda_{U_i} (i = 1, 2, \ldots, M_I - 1)$ under FF is given by $N_2^i(t) = 0$ and the upper barrier by $N_2^i(t) = L + 1$. The random variable Λ_{M_I} under FF is again phase type distributed and represents the time until absorption to a single barrier, namely, $N_2^{M_I}(t) = 0$, starting always from the state $N_1^{M_I}(t) = (M_I - 1)L$, $N_2^{M_I}(t) = 1$.

The initial probability vector (of the phase type distribution) is different under FF for Λ_{D_i} and Λ_{U_i} for $i = 1, 2, \ldots, M_I - 1$. By definition, the random variable Λ_{d_i} is the time until absorption given that the process $\{S_t, t \geq 0\}$ enters state i from state $i-1$. Notice that under FF, entering state i from state $i-1$ will occur if and only if $n_1^i = L(i-1)$ and $N_2^i = 0$. Therefore, the initial probability vector, namely α_{D_i}, takes the value 1 for this state and 0 for all the other states for the case of Λ_{D_i}. On the other hand, for the case of Λ_{U_i}, which is the time until absorption given that the process $\{s-t, t \geq 0\}$ enters state i from state $i+1$, the initial probability vector, namely α_{U_i} corresponds to the joint steady state probability distribution of the number of busy isochronous octets within slots 1 through $i-1$ and within slot i.

This joint steady state probability distribution can be obtained by a brute force solution employing the successive overrelaxation method [6] of a set of steady state equations which can be formulated as a special case $(n = M_I - 1)$ of equations 5 through 12 in Zukerman [34].

The matrix T (in the notation of Neuts [24]) is the same for Λ_{D_i} and Λ_{U_i} (i.e., $T_{D_i} = T_{U_i}$) and based on the infinitesimal generator of the process $\{N_1^i, N_2^i\} = \{N_1^i(t), N_2^i(t), t \geq 0\}$.

If L and M are small, the steady state probability distribution of the delay and packet queue size are obtainable using the above described method. Unfortunately, however, for practical situations where L is over 30 and M is at least 4 (especially under FF where the number of phase states is large) approximations are required. The approximation is performed by modelling a phase type distribution of high order by a phase type distribution of lower order using moment matching. Notice that the exponential system approximation is a special case of this method where it is assumed that the time spent in each capacity mode is exponentially distributed, and only the first moment is matched. As mentioned in [22], in many cases it is essential, especially for the overload modes, to consider more than one moment, and a phase type approximation of at least two phase states is required.

Acknowledgements

The author would like to thank D. Everitt and A. Kennington for reviewing the manuscript and for providing fruitful comments, H. Heffes for suggesting the phase type approximation, and the anonymous reviewers for their constructive suggestions. Also, the permission of the Executive General Manager, Research of Telecom Australia to publish this paper is hereby acknowledged.

References

[1] Z. L. Budrikis, J. L. Hullett, R. M. Newman, D. Economou, F. M. Fosdar and R. D. Jeffery, "QPSX: A queued packet and synchronous circuit exchange", *Proc. ICCC*, Munich (1986).

[2] Z. L. Budrikis, P. A. Evans and R. M. Newman, "QPSX: a proposal from Australia for MAN standard", *IEEE 802 Committee Tutorial Presentation* (1986).

[3] Z. L. Budrikis, J. L. Hullett, R. M. Neman, D. Economou, F. M. Fosdar and J. D. Jeffery, "QPSX: A queued packet and synchronous exchange as a local or metropolitan area network", *Proc. Second Australian Computer Engineering Conference*, Sydney (1986).

[4] K. Caves and A. V. Flatman, "FDDI-II: a new standard for integrated services high speed LANs", *Proc. Wideband Communications*, (1986) 81–91.

[5] E. G. coffman, T. T. Kadota and L. A. Shepp, "A stochastic model of fragmentation in dynamic storage allocation", *SIAM J. Comput.* **14** (1985).

[6] R. B. Cooper, *Introduction to queueing theory* (North Holland, New York, 1981).

[7] P. A. Evans, M. D. Hayes and P. G. Potter, "Metropolitan area networks", *Proc. IREECON'87*, Sydney (1987).

[8] D. C. Feldmeier, *Empirical analysis of a token ring network* Tech. Memo. TM-254, MIT Lab. for Comp. Sci. (1984).

[9] M. J. Fischer and T. C. Harris, "A model for evaluating the performance of an integrated circuite and packet-switched multiplex structure", *IEEE Trans. Comm.* COM-**24** (1976).

[10] E. Gelenbe and C. Rosenberg, *Queues with slowly varying arrival and service process* Report No. 059, Laboratoire I.S.E.M., Universite de Paris-Sud, France (1987).
[11] H. Heffes, "A class of data traffic processes-covariance function characterization and related queueing results", *Bell Syst. Tech. J.* **59** (1980) 897–929.
[12] H. Heffes and D. M. Lucantoni, "A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance", *IEEE J-SAC.* SAC-4 (1986) 856–868.
[13] M. N. Huber, K. Sauer and W. Schodl, "QPSX and FDDI-II performance study of high speed LAN's", *EFOC/LAN* Amsterdam (1988).
[14] J. L. Hullett and P. Evans, "New proposal extends the reach of metro area nets", *Data Communications* (1988).
[15] IEEE 802.6 Committee, *Draft of proposed IEEE standard 802.6 distributed queue dual bus (DQDB) metropolitan area network (MAN)-Draft D6-88/105*, an internal working document of the IEEE 802.6 working group (1988).
[16] D. E. Knuth, *The art of computer programming, volume 2: computer algorithms* (Addison-Wesley, Massachusetts, 1973).
[17] A. G. Konheim and M. Reiser, "The movable-boudnary multiplexor stability and decomposability", *Teletraffic Analysis and Computer Performance Evaluation* (North Holland, 1986).
[18] A. Leon-Garcia, R. H. Kwong and G. F. Williams, "Performance evaluation methods for an integrated voice/data link", *IEEE Trans Comm.* COM-**30** (1982) 1848–1858.
[19] M. Le and L. Neitzel, "FDDI-II working paper", ANSC Document No. X3T9.5 (1986).
[20] D. R. Manfield, "Transport delays in a hybrid switching system with polling", *Proc. IEEE ICC '85*, Chicago, 1985.
[21] K. S. Meier-Hellstern, "A fitting algorithm for Markov-modulated Poisson process having two arrival rates", *European J. of Operational Research* **29** (1987) 370–377.
[22] M. F. Neuts, "The M/M/1 queue with randomly varying arrival and service rates", *Opsearch* **15** (1978) 139–157.
[23] M. F. Neuts, "Further results on the M/M/1 queue with randomly varying rates", *Opsearch* **15** (1978) 158–168.
[24] M. F. Neuts, *Matrix-Geometric solutions in stochastic models: an algorithmic approach* (The John Hopkins University Press, Baltimore, 1981).
[25] R. M. Newman, Z. L. Budrikis and J. L. Hullett, "The QPSX Man", *IEEE Communications Magazine* **26** (1988) 20–28.
[26] R. M. Newman and J. L. Hullett, "Distributed queueing: a fast and efficient packet access protocol for QPSX", *Proc. ICCC '86*, Munich, 1986.
[27] P. O'Reilly, "Performance analysis of data in burst switching", *IEEE Trans. on Commun.* COM-**43** (1986).
[28] V. Ramaswami and D. M. Lucantoni, "Stationary waiting time distribution in queues with pase type service a in quasi birth and death processes", *Commun. Statist. Stochastic Models* **1** (1985) 125–136.
[29] M. H. Rossiter, "A switched Poisson model for data traffic", *Australian Telecommunication Research* **21** (1987).
[30] J. F. Shoch and J. A. Hupp, "Measured performance of an ethernet local network", J. *ACM* **23** (1980) 711–721.
[31] K. Sriram, P. K. Varshney and J. G. Shanthikumar, "Discrete-time analysis of integrated voice/data multiplexers with and without speech activity detectors", *IEEE J-SAC.* SAC-1 6 (1986) 1124–1132.
[32] G. F. Williams and A. Leon-Garcia, "Performance analysis of integrated voice and data hybrid-switched links", *IEEE Trans. Comm.* COM-**32** (1984) 679–706.
[33] U. Yechiali and P. Naor, "Queueing problems with heterogeneous arrivals and service", *Opns. Res.* **19** (1971) 722–734.

[34] M. Zukerman, "On packet switching capacity in QPSX", *Proc. GLOBECOM*, Tokyo (1987).
[35] M. Zukerman, "Circuit allocation and overload control in hybrid switching system", *Computer Networks and ISDN Systems* **16** (1989) 281–298.
[36] M. Zukerman, "Bandwidth allocation for bursty isochronous traffic in a hybrid switching system", to appear in *IEEE Trans. Commun.*
[37] M. Zukerman, "Approximations for performance evaluation of the packet access queue in QPSX: the IEEE 802.6 evolving MAN standard", *A.T.R.* **22** (1988) 53–62.
[38] M. Zukerman and P. Kirton, "Applications of matrix-geometric solutions to the analysis of the bursty data queue in a B-ISDN switching system", *Proc. GLOBECOM '88* **3** (1988) 1635–1639.