Copyright © 1981 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 60, No. 8, October 1981
*Printed in U.S.A.*

# Priority Queuing Networks

By R. J. T. MORRIS

(Manuscript received October 17, 1980)

*Priority service disciplines are widely used in computer and communications systems. Many such systems can be modeled by queuing networks, but presently developed theory does not allow solution of these models when priority service disciplines are present. For priority queuing networks that have a homogeneity property, we give some explicit results for mean delay and throughput. However, the assumption of homogeneity is too restrictive for many applications. We identify some examples of systems for which inhomogeneous two-node priority queuing networks are appropriate models and yield to exact analysis. The results allow some conclusions to be drawn about using priorities in a two-node closed network to establish grades of service. We also use the results to evaluate a commonly used approximation technique for priority queuing systems.*

## I. INTRODUCTION AND SUMMARY

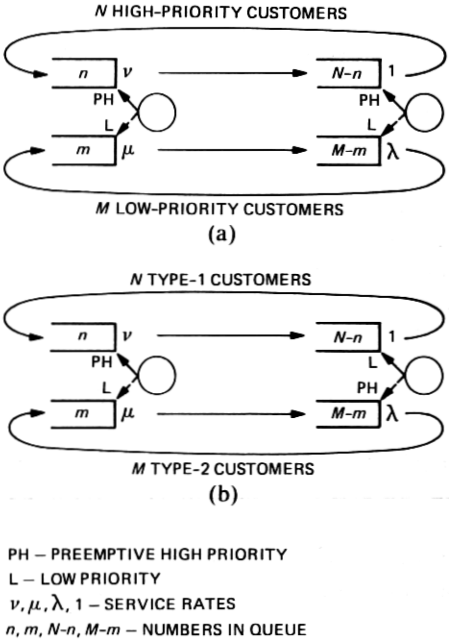
Priority service disciplines are widely used in computer and communication systems. One common application of priorities is in the establishment of multiple grades of service whereby deferrable or background work is scheduled according to a lower priority. In other applications, a device may give prioritized service to a class of jobs known to be short so as to increase overall system throughput. For purposes of performance analysis, computer and communication systems have often been modeled as queuing networks. However, the theory of queuing networks in its present form (see Refs. 1, 2) does not provide solutions for even simple networks with priority disciplines, except in an approximate manner.[3–5]

There are very few exact results for queuing networks (i.e., queuing models with more than one service station or node) with priority to be found in the literature. One known result concerns a general service time, single-server queue with preemptive or nonpreemptive priority and finite exponential source.[6] This model can be thought of as a two-node closed queuing network where the second node is a pure delay or infinite-server group. In a paper by Avi-Itzhak and Heyman the mean cycle times were obtained for a central server model with priorities, under the assumption that the mean service times and routing patterns are the same for each priority class.[7] In other computer and communication applications, network priorities have only been represented approximately, using heuristics for the central server model[3,4] and a packet switching network.[5] While these approximation techniques may be adequate in accuracy for the parameter ranges of some applications, much further work is needed in improving and validating these techniques and, ultimately, in developing exact analytical results wherever they are tractable.

Our goals are to obtain insight into the solution form for some simple cases of queuing networks with priorities, to obtain an initial evaluation of the accuracy of existing approximation techniques, and to draw some conclusions on the performance of some simple network priority structures occurring in practice. In Section II, we describe a general class of priority queuing networks that are homogeneous in the sense that all customer classes are treated identically with respect to service time and routing. For homogeneous networks, we are able to give a mean delay and throughput analysis. However, it will be seen that the homogeneity assumption is sufficiently restrictive as to prevent application of these results in many situations. Subsequently, we focus on two specific examples of systems that can be modeled by simple queuing networks, but in which priority disciplines and inhomogeneity play crucial roles. These examples suggest several different two-node priority queuing network models that yield to exact analysis.

The first example we consider is a computer system consisting of a central processing unit (CPU) and an input/output (I/O) device, which processes both time-critical transactions, as well as nontime-critical batch jobs. The system is designed to give priority to the transactions at both the CPU and I/O device (in contrast to Refs. 3 and 4 where it is assumed that only the CPU observes priorities). This suggests use of the two-node, closed queuing network model A of Fig. 1a, with one node representing the CPU, and the other, the I/O device. The model has separate queues for each priority class at each node, and priority is observed preemptively at both nodes.

The second example is a full-duplex data link which is used for transmission of messages under a window flow control protocol. There are two grades of messages, the premium grade and the standard grade. When both premium grade messages and acknowledgments receive preemptive priority, model A is applicable (refer to Fig. 5 which is explained further in Section VI). However, since acknowledgments are typically shorter than messages, another configuration is


N HIGH-PRIORITY CUSTOMERS
n
ν
N-n
1
PH
L
PH
L
m
μ
M-m
λ
M LOW-PRIORITY CUSTOMERS
N TYPE-1 CUSTOMERS
n
ν
N-n
1
PH
L
L
PH
m
μ
M-m
λ
M TYPE-2 CUSTOMERS
PH – PREEMPTIVE HIGH PRIORITY
L – LOW PRIORITY
ν, μ, λ, 1 – SERVICE RATES
n, m, N-n, M-m – NUMBERS IN QUEUE

(a)

(b)

Fig. 1—Schematics of models A and B.

suggested wherein acknowledgments of either grade are given preemptive priority; this leads to model B shown in Fig. 1b.

In both models A and B there are a fixed number of customers in each class and service time distributions at a node are assumed to be exponential, but are not required to be the same at a node for each customer class (in contrast with homogeneous networks or the first-come first-served nodes described in Ref. 1). Because of the exponential assumption, the priorities can be understood to be either preemptive-resume or preemptive-restart (with resampling). For each model, service within a customer class at a node can be thought of as first-come first-served, but all equations and results remain valid for any other discipline within the priority class which does not take into account the actual service time requirement when selecting a customer for service.

The general approach we use in the analysis of models A and B, and several similar models, is to set up the balance equations (steady-state Kolmogorov forward equations) for the Markov chain describing the number of customers of each priority class at each node. These partial difference equations generally do not satisfy the well-known local balance condition[1,2] but, nevertheless, can still be solved to obtain the stationary distribution. This distribution allows throughput and mean delay to be computed for each customer class.

These results can be applied to obtain some general conclusions about the two systems we have used as motivating examples. In the computer system that processes both transactions and batch jobs, we find that if the transactions are bottlenecked at one device (CPU or I/O), the batch jobs need to be even more strongly bottlenecked at the other device, if a significant batch throughput is to be attained. Specifically, we show that if the transactions have a bottleneck of strength $x$ at one node, the batch jobs need to have a bottleneck of strength $x^N$ at the other node (where $N$ is the transaction multiprogramming level), if the batch jobs are to be able to fulfill a role as "filler" work. In the data link example, we find a similar result: if standard grade message traffic is carried in purely a background mode, fairly extreme parameters are necessary before its introduction becomes attractive. On the other hand, if some compromise of the premium traffic performance is permitted, then an appreciable amount of standard grade message traffic can be carried by using data link capacity that would otherwise be wasted. For each system, we identify hazards that can occur when the lower-priority work is allowed to interfere with higher-priority work. Refer to Sections V and VI for further details.

In Section VII, we use the results to evaluate the effectiveness of a well-known approximation technique. We find that accuracy of the approximation technique varies from good to poor, depending on the parameters of the application. A criterion on the application parameters is proposed under which the approximation technique would be expected to perform well.

## II. HOMOGENEOUS NETWORKS

In this section, we consider a class of queuing networks that allow preemptive priorities but are otherwise homogeneous in the sense that at any one node all customers are treated identically with respect to service rate and routing. The results rely on an observation similar to that made by Avi-Itzhak and Heyman in their analysis of the central server model.[7]

We first consider a closed queuing network of the Gordon-Newell type.[8] It consists of $N$ service centers or nodes numbered $1, \cdots, N$. In departure from the Gordon-Newell formulation, there are $P$ priority classes numbered $1, \cdots, P$, with the $i$th (priority) class containing $K_i$ customers, $i = 1, \cdots, P$ and

$$\sum_{i=1}^{P} K_i = K.$$

At any node, a customer from a higher-numbered class takes preemptive priority over a customer from a lower-numbered class. The service time distribution at node $j$ is exponential with rate $\mu_j$ for all customers, and the service discipline is first-come first-served within each priority class. After service at a node is completed, routing to another node is governed by a probability vector which is the same for each priority class. Let the state of the network be expressed by the quantities $n_j^i$, $j = 1, \cdots, N$, $i = 1, \cdots, P$, where $n_j^i$ denotes the number of class $i$ customers present at node $j$. Define the aggregate state variable

$$m_j^i = \sum_{k=i}^{P} n_j^i,$$

the number of priority class $i$ or higher customers at node $j$. The key observation is that the random variable $m_j^i$ is equivalent to that which would result if the network were modified by first removing customers of priority less than $i$ (i.e., by setting $K_k = 0, 1 \leq k < i$) and, thereafter, ignoring all priority service distinctions. This is because (*i*) lower-priority customers exert no influence on higher-priority customers, and (*ii*) regardless of whether priorities are observed between customers of classes $i, i+1, \cdots, P$, transitions in the total number ($m_j^i$) of customers of priority $i$, or larger, at a node are not altered (by the assumed uniformity of service rate and routing over priority classes). Thus, we can find the stationary distribution of $m_j^i$ by the usual closed queuing network techniques.[1,2,8,9] The stationary distribution of the aggregate variable $m_j^i$ is sufficient to determine the steady-state mean delay and throughput of each priority class at each node. This follows from the fact that for each $i$ and $j$

$$E[n_j^i] = E[m_j^i] - E[m_j^{i+1}],$$

$$\Pr[n_j^i > 0, m_j^{i+1} = 0] = \Pr[m_j^i > 0] - \Pr[m_j^{i+1} > 0],$$

where $m_j^{P+1}$ is understood to be identically zero. Hence, priority class $i$ customers have a throughput at node $j$ of

$$T_j^i = \mu_j[\Pr(m_j^i > 0) - \Pr(m_j^{i+1} > 0)]$$

and a mean delay (including service time) of

$$D_j^i = [E(m_j^i) - E(m_j^{i+1})]/T_j^i$$

by Little's Law. Note that these quantities are obtained in the process of carrying out the mean value analysis for a single-chain closed network with $K$ customers.[9]

Similar results are obtained for an open network of the Jackson type.[10] All notation is the same as for the closed network, except that we no longer specify the number of customers of each priority class but, instead, we specify the rate $\lambda_j^i$ of exogenous Poisson arrivals of priority class $i$ to node $j$. We must now assume that the traffic equations admit a unique solution $e_j^i$ representing the mean arrival rate of customers of priority class $i$ to node $j$ and that

$$\sum_{i=1}^{P} e_j^i < \mu_j$$

for each $j$. We make the analogous observation that the quantity $m_j^i$ can be obtained by considering the network modified by turning off all arrival streams of priority less than $i$ (i.e., setting $\lambda_j^k = 0$, $1 \le k < i$, $1 \le j \le N$) and, thereafter, ignoring priority distinctions at service. We then have

$$E[n_j^i] = E[m_j^i] - E[m_j^{i+1}],$$
$$T_j^i = e_j^i,$$

and

$$D_j^i = [E(m_j^i) - E(m_j^{i+1})]/e_j^i.$$

Now,

$$E[m_j^i] = \sum_{k=i}^{P} e_j^k \Big/ \left(\mu_j - \sum_{k=i}^{P} e_j^k\right)$$

and, therefore,

$$D_j^i = \frac{\mu_j^{-1}}{\left(1 - \sum_{k=i}^{P} \rho_j^k\right)\left(1 - \sum_{k=i+1}^{P} \rho_j^k\right)},$$

where $\rho_j^k = e_j^k/\mu_j$ is the utilization of node $j$ due to class $k$ customers. We, thus, recognize the validity in a network context of the Cobham-type formula originally obtained by White and Christie[11] for the delay in an isolated preemptive priority $M/M/1$ queue when the service times are the same for each priority class.

Within the stringent limitations imposed by our homogeneity assumptions, some extensions to these results are possible. For example, we can allow the more general service disciplines shown by Kelly (see Ref. 2, pages 58 and 78) to lead to product form provided (*i*) the state dependence and server sharing embodied in these disciplines extend only to the customers of highest priority present at a node and ignore all lower-priority customers, and (*ii*) all customers are treated identically with respect to service time and routing.

The above results might possibly be useful in some queuing network applications. For example, a first-cut evaluation of the impact of introducing data packet priorities into a packet switching network could be carried out by representing exogenous packet arrivals as Poisson and data links as exponential servers,[12,13] and by assuming that the mean data packet length and the traffic routing pattern are the same for each priority class. If short control (e.g., acknowledgment) packets were to be given priority over data packets, we find that our homogeneity assumptions would be violated, although the effect of the short packets could be approximated in several ways.[5,13] Indeed, the case of control packets receiving priority serves to illustrate a common situation in which customers receiving priority have significantly smaller service time requirements. Thus, the results described in this section are expected to find limited use. The remainder of this paper does not make any such homogeneity assumption; unfortunately, by relaxing this assumption, we are able to treat only networks consisting of two nodes.

## III. MODEL A: TWO NODES WITH PRIORITIES THE SAME AT EACH NODE

We now consider the two-node closed queuing network introduced in Section I as model A and shown in Fig. 1a. The network consists of two nodes—the left- and right-hand nodes. There are $N$ high-priority and $M$ low-priority customers. High-priority customers take preemptive priority over low-priority customers at each node. All service times are assumed exponentially distributed: the high-priority customers have a mean service time of $\nu^{-1}$ at the left node and 1 at the right; low-priority customers have a mean service time of $\mu^{-1}$ at the left node and $\lambda^{-1}$ at the right. After service at one node is completed, customers are immediately routed to the other node without changing class. We assume $\nu$, $\mu$, $\lambda$, $N$, and $M$ are all positive.

The state of the system is described by the vector $(n, m)$ where $n$ (respectively $m$) is the number of high- (respectively low) priority customers at the left node. The state $(n, m)$ evolves as a Markov chain with stationary distribution $p(n, m)$. It is obvious that the stationary distribution is also the limiting distribution since the chain is finite and irreducible. The transitions of $(n, m)$ are shown in Fig. 2a.

By definition, $p(n, m)$ satisfies the balance equations

$$\begin{aligned} p(n, m)[\nu 1_{\{n>0\}} + \mu 1_{\{n=0,m>0\}} + 1_{\{n<N\}} + \lambda 1_{\{n=N,m<M\}}] \\ = p(n-1, m) + p(n+1, m)\nu + p(N, m-1)\lambda 1_{\{n=N\}} \\ + p(0, m+1)\mu 1_{\{n=0\}}, \qquad 0 \le n \le N, \qquad 0 \le m \le M, \quad (1) \end{aligned}$$

where $1_{\{\ \}}$ denotes the indicator function which has value 1 (respectively 0) when the predicate within the braces is true (respectively false). Note that we are adopting the convention that $p(n, m) = 0$ when $(n, m) \notin [0, N] \times [0, M]$.

We wish to solve for $p(n, m)$. The technique we use is best explained by reference to the state transition diagram shown in Fig. 2a. First

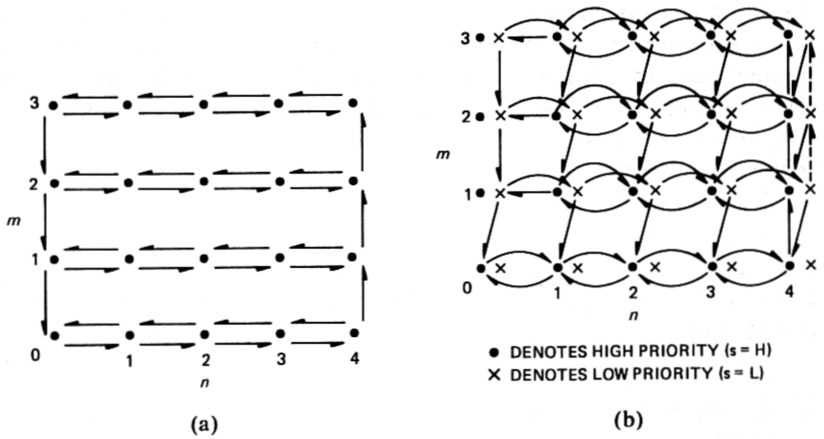

3
2
m
1
0
1
2
3
4
n
(a)
3
2
m
1
0
1
2
3
4
n
● DENOTES HIGH PRIORITY (s = H)
× DENOTES LOW PRIORITY (s = L)
(b)

Fig. 2—(a) State transition diagram for model A shown for N = 4, $M = 3$. (b) State transition diagram for model A, but with left node nonpreemptive, shown for $N = 4$, $M = 3$.

note that for any $n > 0$, $p(n, m)$, $0 \le m \le M$ is expressible in terms of $p(n-1, m)$, $0 \le m \le M$. Hence, $p(n, m)$, $0 \le m \le M$, $0 < n \le N$ can be expressed in terms of the left boundary values $p(0, m)$, $0 \le m \le M$, and solution for $p(0, m)$, $0 \le m \le M$ rests on the balance equations for the right boundary $p(N, m)$, $0 \le m \le M$. This observation is generally true for an arbitrary two-dimensional birth-death process (provided all right-to-left horizontal transitions are present) but not always useful since the resultant equations for $p(0, m)$, $0 \le m \le M$ are not easily solved. Fortunately, in our case, the absence of vertical transitions from states $(n, m)$, $0 < n < N$ caused by the priority structure results in relations for $p(0, m)$ which comprise a simple difference equation of order two which is easily solved and yields an explicit solution for $p(n, m)$. Before proceeding with this technique, we mention that a computational method based on such an observation has been proposed in which the recursive structure is used to reduce the problem of finding the stationary distribution of certain $N \times M$ birth-death processes to the solution of $N$ equations in $N$ unknowns.[14]

For notational ease, define $\alpha(m) = p(0, m)$, $0 \le m \le M$. Writing eq. (1) for $n = 0$ yields

$$p(1, 0) = \alpha(0)\nu^{-1} - \alpha(1)\mu\nu^{-1}, \tag{2}$$

$$p(1, m) = \alpha(m)(\mu + 1)\nu^{-1} - \alpha(m + 1)\mu\nu^{-1}, \qquad 0 < m < M \tag{3}$$

$$p(1, M) = \alpha(M)(\mu + 1)\nu^{-1}, \tag{4}$$

and for $0 < n < N$

$$p(n, m)(\nu + 1) = p(n + 1, m)\nu + p(n - 1, m), \qquad 0 \le m \le M \tag{5}$$

for which the general solution is

$$p(n, m) = (\alpha(m)(\nu^{-n} - \nu^{-1}) + p(1, m)(1 - \nu^{-n}))/(1 - \nu^{-1}),$$
$$0 \le n \le N, \qquad 0 \le m \le M, \tag{6}$$

provided $\nu \neq 1$. Hence, the problem is reduced to determination of $\alpha(m)$, $0 \le m \le M$. This is done by writing eq. (1) for $n = N$,

$$p(N, 0)(\nu + \lambda) = p(N - 1, 0), \tag{7}$$

$$p(N, m)(\nu + \lambda) = p(N - 1, m) + p(N, m - 1)\lambda, \qquad 0 < m < M, \tag{8}$$

$$p(N, M)\nu = p(N - 1, M) + p(N, M - 1)\lambda. \tag{9}$$

Substituting eqs. (2) and (6) into eq. (7),

$$\alpha(1)/\alpha(0) = (\lambda/\mu)[\nu^{-N}(\nu - 1)]/(\nu + \lambda - 1 - \lambda\nu^{-N}). \tag{10}$$

Assuming $M > 1$, taking $m = 1$ in eq. (8), and using eqs. (2), (3), and (6) yields $\alpha(2)/\alpha(1) = r$, where

$$r = \frac{\lambda}{\mu}\frac{\mu - (\mu - \nu + 1)\nu^{-N}}{\nu + \lambda - 1 - \lambda\nu^{-N}}. \tag{11}$$

Note that the denominator of $r$ is not zero because $\nu \neq 1$. Taking $1 < m < M$ in eq. (8) with eqs. (3) and (6) yields the difference equation

$$\alpha(m + 1)[\mu\lambda\nu^{-N} + \mu - \mu\lambda - \mu\nu]$$
$$+ \alpha(m)[\mu\nu + \lambda\nu^{1-N} + 2\lambda\mu - 2\lambda\mu\nu^{-N} - \lambda\nu^{-N} - \mu]$$
$$+ \alpha(m - 1)[\lambda\mu\nu^{-N} + \lambda\nu^{-N} - \lambda\mu - \lambda\nu^{1-N}] = 0, \qquad 1 < m < M,$$

which has characteristic roots 1, $r$. But since $\alpha(2)/\alpha(1) = r$, we have

$$\alpha(m) = \alpha(1)r^{m-1}, \qquad 1 \le m \le M. \tag{12}$$

It can now be verified that these results are consistent with the one unused eq. (9) and that the result holds true for $M = 1$.

Substituting eqs. (2) to (4), (10), and (12) into eq. (6) and simplifying yields the general solution for $\nu \neq 1$

$$p(n, m) = Cr^m[r\nu^{-n\delta(m-M)} - \nu^{(N-n)\delta(m)} + ((1 - \nu)\mu^{-1}$$
$$+ 1 - r)\nu^{-n}], \qquad 0 \le n \le N, \qquad 0 \le m \le M, \tag{13}$$

where $r$ is given by eq. (11) and $\delta(\cdot)$ is the Kronecker delta: $\delta(0) = 1$; $\delta(k) = 0$, $k \neq 0$. The normalizing constant $C$ is obtained by demanding that $p(n, m)$, $0 \le n \le N$, $0 < m \le M$ is a probability distribution, yielding

$$C = \frac{\mu(1 - \nu^{-1})}{(1 - \nu^{-N-1})\left[(1 - \nu)\sum_{i=0}^{M} r^i + \mu(1 - \nu^N)\right]} \tag{14}$$

In eq. (14) and below we refrain from summing geometric series of the form

$$\sum_{i=0}^{I} x^{i}$$

to avoid a special statement for the case $x = 1$.

Our treatment has excluded the case $\nu = 1$ for which the solution to eq. (5) is $p(n, m) = \alpha(m)(1 - n) + p(1, m)n$. Rather than resolving for this case, it is easier to take the limit as $\nu \rightarrow 1$ in eqs. (13) and (14); this must yield the correct solution since the coefficients in eq. (1) are continuous in $\nu$ and the eigenvectors of a matrix are continuous functions of its elements. Equations (13) and (14) for $p(n, m)$ can be rewritten in a form which is well-defined for $\nu = 1$:

$$p(n, m) = Ds^{m}\left[\mu^{-1}\nu^{-n} + (1 - s)\nu^{-1}\sum_{i=0}^{n-1}\nu^{-i} + 1_{\{m=0\}}\sum_{i=0}^{N-n-1}\nu^{i} + 1_{\{m=M\}}s\nu^{-1}\sum_{i=0}^{n-1}\nu^{-i}\right], \qquad (15)$$

where

$$s = \left(\sum_{i=0}^{N-1}\nu^{-i} + \mu^{-1}\nu^{1-N}\right)\Big/\left(\sum_{i=0}^{N-1}\nu^{-i} + \lambda^{-1}\nu\right), \qquad (16)$$

$$D^{-1} = \left(\sum_{i=0}^{N-1}\nu^{i} + \mu^{-1}\sum_{i=0}^{M}s^{i}\right)\sum_{i=0}^{N}\nu^{-i}, \qquad (17)$$

and summations over descending ranges are taken to be zero. Since the solution given by eqs. (15) to (17) is continuous in $\nu > 0$, it is the general solution for all $\nu > 0$.

### Remarks

(*i*) The system considered here can be generalized trivially to allow mean service time of the high-priority customers at the right node to be $\kappa^{-1}$ (rather than unity) and to allow routing of a customer back to the node at which it has just completed. Let high-priority customers on completion at the left (respectively right) node be routed to the right (respectively left) node with probability $p_{lr}$ (respectively $p_{rl}$) and be routed to the node at which completion has just occurred with complementary probability $1 - p_{lr}$ (respectively $1 - p_{rl}$). Let the low-priority customers have similarly defined probabilities $q_{lr}$, $q_{rl}$. Then the results in eqs. (13) to (17) remain valid with $\nu$, $\mu$, $\lambda$ replaced, respectively, by $\nu p_{lr}/\kappa p_{lr}$, $\mu q_{lr}/\kappa p_{rl}$, $\lambda q_{rl}/\kappa p_{rl}$.

(*ii*) It is readily verified that the marginal distribution of the high-priority customers $p(n, \cdot) = \sum_{m=0}^{M} p(n, m)$ agrees with the result for an ordinary two-node closed network, viz.

$$p(n, \cdot) = \nu^{-n} \bigg/ \sum_{i=0}^{N} \nu^{-i}.$$

Of course, this must be the case, since the high-priority customers experience no interference from the low-priority customers.

(*iii*) Let $T_H$ and $T_L$ denote the throughput of high- and low-priority customers, respectively. Then

$$T_H = \nu(1 - p(0, \cdot)) = \nu\left[1 - \left(\sum_{i=0}^{N} \nu^{-i}\right)^{-1}\right] \quad \text{and}$$

$$\begin{aligned} T_L &= \mu[p(0, \cdot) - p(0, 0)] \\ &= \sum_{i=1}^{M} s^i \bigg/ \left(\sum_{i=0}^{N-1} \nu^i + \mu^{-1} \sum_{i=0}^{M} s^i\right) \sum_{i=0}^{N} \nu^{-i}. \end{aligned}$$

If the generalization referred to in (*i*) is adopted, then as well as the replacements specified in (*i*), the quantities $T_H$, $T_L$ must be multiplied by $\kappa p_{rl}$ if they are to have the units of customers per unit time.

Mean delay formulas for high- and low-priority customers at each node are immediately obtained from $p(n, m)$ by Little's law, but are omitted here.

(*iv*) In the special case that $\mu = \nu$ and $\lambda = 1$, i.e., service times do not depend on the priority class, the throughputs can be obtained from the considerations of Section II. In that case, the distribution of $n + m$ will be the same as a two-node closed network with $N + M$ customers and no priorities. Hence, we can immediately write down

$$T_H = \nu\left[1 - \left(\sum_{i=0}^{N} \nu^{-i}\right)^{-1}\right],$$

$$\begin{aligned} T_L &= \nu\left[1 - \left(\sum_{i=0}^{N+M} \nu^{-i}\right)^{-1}\right] - T_H \\ &= \nu\left[\left(\sum_{i=0}^{N} \nu^{-i}\right)^{-1} - \left(\sum_{i=0}^{N+M} \nu^{-i}\right)^{-1}\right], \end{aligned}$$

and this checks with the result in (*iii*).

Note also that in this case ($\mu = \nu$, $\lambda = 1$), then $s = \nu^{-1}$ and $p(n, m) = D\nu^{-1-m}$, $0 < m < M$. Thus, if we observe the system only at instants when there is at least one low-priority customer at each node, the distribution of high-priority customers is seen to be uniform.

(*v*) When $s \geq 1$, using the expressions in (*iii*)

$$\lim_{M\to\infty} \frac{T_H}{\nu} + \frac{T_L}{\mu} = 1,$$

i.e., the utilization of the left-hand node approaches unity as the number of low-priority customers increases. It follows from the definition of $s$ that $s \geq 1$ if and only if $\lambda/\mu \geq \nu^N$. Using this fact and reversing the two nodes shows conversely that if $s \leq 1$, utilization of the right-hand node approaches unity as $M \to \infty$. Thus, whether $s > 1$ or $s < 1$ determines which node becomes the limiting factor in trying to obtain increased total throughput by introducing additional low-priority customers.

The following criterion can be deduced. Suppose a two-node system with $N$ circulating customers has a bottleneck of strength $\nu$, $\nu > 1$ at the right node. For moderate values of $N$, the right node will be almost completely utilized and the left node underutilized. If the low-priority customers have a bottleneck at the left node of strength at least $\nu^N$, i.e., $\lambda/\mu \geq \nu^N$, then the left node can have a utilization as close as desired to unity by introduction of sufficiently many low-priority customers. If the low-priority customers have a bottleneck weaker than $\nu^N$ at the left node, then complete use of the left node can never be achieved by introducing low-priority customers. This rule of thumb can be deduced intuitively as follows. The high-priority customers can be thought of as causing a reduction of processing rate to $\mu[1 - T_H/\nu]$ at the left node and to $\lambda[1 - T_H]$ at the right node. Thus, the left node can be fully utilized if and only if $\mu(1 - T_H/\nu) \leq \lambda(1 - T_H)$ which reduces to the condition $\lambda/\mu \geq \nu^N$. Of course, the formulas for $T_H$ and $T_L$ make precise the actual throughputs achieved as a function of the parameters. This is illustrated in Section V by an example.

(*vi*) The same solution technique can be used to obtain results for more than two priority classes, although the solution complexity increases. We state the result for a three-class, two-node system with number in each class $N$, $M$, $L$ (in order of decreasing priority), with service time at the left node $\mu^{-1}$ (for all classes) and 1 at the right, $\mu \neq 1$. If $p(n, m, l)$ describes the stationary probability of having $n$, $m$, $l$ customers at the left (in order of decreasing priority), then, provided $N$, $M$, $L$, $\mu$ are positive,

$$p(n, m, l) = \begin{cases} b(\mu^{-n} - \mu^{-1-N}), & l = 0, \quad m = 0 \\ B\mu^{-l}(\mu^{-n} - \mu^{-1-N}), & 0 < l \leq L, \quad m = 0 \\ B\mu^{M-m} - B\mu^{-1-n}, & l = 0, \quad 0 < m < M \\ B(1 - \mu^{-1})\mu^{-l-n}, & 0 < l < L, \quad 0 < m < M, \\ B\mu^{-L}(\mu^{-n} - \mu^{-N-m-1}), & l = L, \quad 0 < m < M \\ B\mu^{-l}(1 - \mu^{-n-1}), & 0 \leq l < L, \quad m = M \\ b\mu^{-M-N-L}(1 - \mu^{-n-1}), & l = L, \quad m = M \end{cases}$$

where

$$B = b(1 - \mu)/(1 - \mu^{M+N+1})$$

and

$$b = (1 - \mu^{-1})/[(1 - \mu^{-N-1})(1 - \mu^{-M-N-L-1})].$$

A result for three priority classes is sometimes useful in that it allows comparison of the performance of a designated customer class with two aggregated classes: one representing those customer classes of higher priority and the other those customer classes of lower priority.

(*vii*) A variation on model A which will be of interest in Section VI is the case where priority is again preemptive at the right node but nonpreemptive at the left node. We use the same notation as above, except that the state description now becomes $(n, m, s)$, where $s = H$ (respectively $L$) when the left node is processing high- (respectively low) priority customers. The state transition diagram for this model is shown in Fig. 2b where transitions out of the transient states $\{(n, 0, L), 0 \le n \le N; (0, m, H), 0 < m \le M\}$ are omitted and we have adopted the convention that $s = H$ when $n = m = 0$. The stationary probabilities $p(n, m, s)$ can be solved for by a technique similar to that used above. Namely, letting $p(N, m, L) = \alpha(m)$, $1 \le m \le M$, $p(0, 0, H) = \alpha(0)$ and writing balance equations for all states with $s = L$, yields expressions for $p(n, m, L)$, $1 \le m \le M$, $0 \le n \le N$ and $p(1, m, H)$, $0 \le m \le M$ in terms of $\alpha(\cdot)$. The balance equations for states $(n, m, H)$, $0 \le m \le M$, $1 \le n < N$ yield $p(n, m, H)$, $0 \le m \le M$, $1 \le n \le N$, again in terms of $\alpha(\cdot)$. The analysis is completed by solving the third-order constant coefficient difference equation for $\alpha(\cdot)$ that is obtained from the balance equations for states $(N, m, H)$, $0 \le m \le M$. This approach leads to a solution which, although closed form, is of limited use because of its complexity.

Instead, we now briefly describe a much simpler approximate solution which appears justifiable for our applications. The system is approximated by omitting the transitions shown by dashed lines in Fig. 2b, i.e., $(N, m, L) \rightarrow (N, m + 1, L)$, $0 < m < M$. With these transitions omitted, the solution steps simplify and we obtain

$$p(n, m, L) = \beta(m)(\nu - 1)(\mu + 1)^{-n-1}(1 + \mu^{-1})^{\delta(n-N)}, \qquad 1 \le m \le M, \qquad 0 \le n \le N,$$

$$p(n, m, H) = \beta(m)(1_{\{m>0\}} - \nu^{-n}) + \beta(m + 1)$$
$$\cdot\{-1 + [\mu\nu^{-n} - (\mu + 1)^{-n}(\nu - 1)]/(\mu - \nu + 1)\}1_{\{m < M\}}, \qquad 0 \le m \le M, \qquad 1 \le n \le N \quad \text{or} \quad m = n = 0,$$

where

$$\beta(m) = Ct^{m-1}(1 - \nu^{N})^{\delta(m)},$$

$$t = \frac{\lambda(1 - \nu^{-N})(\mu - \nu + 1)}{(\mu - \nu + 1)(\nu + \lambda - 1) + \lambda(\mu + 1)^{-N}(\nu - 1) - \lambda\mu\nu^{-N}},$$

$C$ is a normalizing constant, $M > 1$, $N > 1$, $\nu \neq 1$, $\nu \neq \mu + 1$, and all unspecified probabilities are zero. As before, the cases $\nu = 1$, $\nu = 1 + \mu$ are obtained by taking limits.

The omission of the dashed transitions amounts to a denial of service to low-priority customers at the right node when $n = N$ and $s = L$. This is a justifiable approximation when the probability is small that $N$ high-priority customers and one low-priority customer can be served at the right in less time than it takes to serve one low-priority customer at the left. This condition is stated as $(1 + \mu)^{-N}\lambda/(\lambda + \mu) \ll 1$, and this expression is shown to hold for our applications in Section VI. The approximation causes an underestimation of low-priority throughput and an overestimation of high-priority throughput. Although we omit details here, the opposite bounds (an upper bound for low-priority throughput and a lower bound for high-priority throughput) could also be obtained by replacing the transition $(N, m, L) \rightarrow (N, m + 1, L)$ by $(N, m, L) \rightarrow (N, M, L)$, $0 < m < M$, leading, in turn, to a system which is solved the same way.

## IV. MODEL B: TWO NODES WITH PRIORITIES REVERSED

Model B, shown in Fig. 1b is now considered. This queuing network differs from model A only in that the priority at the right node has been reversed. We now refer to $N$ type-1 customers and $M$ type-2 customers (see Fig. 1b) with $n$(respectively $m$) being the number of type 1 (respectively type 2) customers at the left node. We assume $N$, $M$, $\nu$, $\mu$, $\lambda$ are all positive.

The state transition diagram for this model is shown in Fig. 3. One immediately recognizes that states $\{(n, m): n > 0 \text{ and } m < M\}$ are transient, i.e., in equilibrium, one of the two high-priority queues is always empty. This is also easily deduced by considering system behavior at instants after the state $(0, M)$, i.e., both high-priority queues empty, is reached. One of the low-priority queues completes a

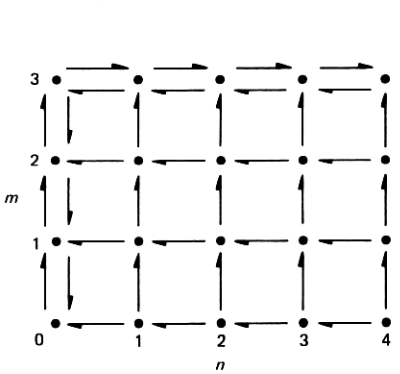


Fig. 3—State transition diagram for model B shown for $N = 4$, $M = 3$.

service and then that customer type prevents service of the other customer type until the state $(0, M)$ is again attained.

Let $p(n, m)$ be the stationary probability of state $(n, m)$. Using the observation as to which states are persistent, we have immediately that

$$p(0, m) = C(\lambda/\mu)^m, \qquad 0 \le m \le M \qquad \text{and}$$
$$p(n, M) = C(\lambda/\mu)^M \nu^{-n}, \qquad 0 \le n \le N,$$

where

$$C^{-1} = \sum_{m=0}^{M=1} (\lambda/\mu)^m + (\lambda/\mu)^M \sum_{n=0}^{N} \nu^{-n}$$

and all other probabilities are zero. The values of $p(n, m)$ are also the limiting probabilities.

***Remarks***

(*i*) For the case $\nu = \mu$, $\lambda = 1$, this result can be obtained directly from standard closed queuing network results, together with the observation regarding persistent states.

(*ii*) For this model, there is no absolute priority given to either type of customer over the other—the service received by each type is determined by parameter values. We can write down the throughput of type 1 (respectively type 2) customers, denoted $T_1$ (respectively $T_2$):

$$T_1 = \nu \sum_{n=1}^{N} p(n, M) = \nu \Big/ \left[1 + \left(\sum_{n=1}^{N} \nu^{-n}\right)^{-1} \sum_{m=0}^{M} (\lambda/\mu)^{-m}\right],$$

$$T_2 = \mu \sum_{m=1}^{M} p(0, m) = \lambda \Big/ \left[1 + \left(\sum_{m=1}^{M} (\lambda/\mu)^{-m}\right)^{-1} \sum_{n=0}^{N} \nu^{-n}\right].$$

(*iii*) In view of the earlier observation regarding transient states, we point out one aspect of the behavior of this system. As already described, in equilibrium, the system will alternate between periods during which only customers of one type are processed. The distribution of the lengths of these periods can be obtained using a standard $M/M/1/K$ ($K$ waiting positions) busy period argument. In the situation that $\nu \ll 1$ (respectively $\lambda \ll \mu$), while each customer type may perceive satisfactory (long-term) *average* throughput, the duration of the period during which only type 1 (respectively type 2) customers are served can be extremely long. The adverse impact of such behavior on the tails of delay distributions is obvious. This phenomenon should be taken into account when an application requires good short-term, as well as long-term performance.

(*iv*) An interesting result for this system is that the delay of one type of customer at its higher-priority queue is not influenced by the presence of customers of the other type. For example, the mean delay of type 1 customers at the left node is given by

$$\sum_{n=1}^{N} n\nu^{-n} \Big/ \sum_{n=0}^{N-1} \nu^{-n}$$

for any $\lambda$, $\mu$, $M$, corresponding to the ordinary closed queuing network result where $M$ would be zero. This invariance is explained as follows. If the random variable $n$ is observed only at instants when it satisfies $n > 0$, then it is indistinguishable from its behavior in an ordinary queuing network where $M$ would be zero. This is because, in equilibrium, $n > 0$ implies $m = M$, and so the type 1 customer sees no interference from type 2 customers. But type 1 customer delays at the left are only measured when $n > 0$ and therefore, the distribution of delay is unaffected by type 2 customers.

(*v*) A variation on model B where one node, say the left, has nonpreemptive priorities can be solved by the same technique. In this case, the only persistent states are $\{(n, m, L): n = 0 \text{ or } m = M\}$ and $\{(n, m, H): n > 0 \text{ and } m \geq M - 1\}$.

## V. COMPUTER SYSTEM EXAMPLE

We now use the results we have developed to evaluate several performance issues in a computer system. We consider a simplified model of a computer system consisting of a CPU and I/O device. The system is primarily intended to process time-critical transactions which it does with a multiprogramming level of $N$. Each transaction makes I/O requests requiring a mean service time of 10 ms separated by CPU processing of mean duration 5 ms. After a certain number of loops between the CPU and I/O device, the transaction is completed and leaves the system. At this point, the transaction is considered to immediately re-enter the system in accordance with the assumption that there is always a backlog of transactions waiting outside the system.*

The transaction workload is clearly I/O-bound and we ask whether introduction of CPU-bound batch "filler" or background work at lower priority will result in a worthwhile improvement of CPU utilization and, consequently, total throughput. Suppose batch jobs are introduced with a permitted multiprogramming level of $M$ and that they require $\gamma$ seconds of processing on the average between visits to the I/O device where mean service time is 10 ms. We again assume that the batch multiprogramming level of $M$ is maintained by a backlog of work.*

* Each transaction or batch job alternately visits the CPU and I/O device, beginning with the CPU, ending with the I/O device and looping between the two an arbitrarily distributed number of times. Variations on this "scenario" can be modeled using the technique in Remark (*i*), Section III.

We will initially assume that the transactions are given preemptive priority at both the CPU and I/O device. This arrangement would reflect an attitude that the performance of high-priority transaction work should not be compromised by introduction of background work. Hence, we use model A, and will be assuming that all service times are exponentially distributed. Because of this exponential assumption, the preemption can be either resume or restart (with resampling). Preemptive-resume is more appropriate at the CPU, and preemptive-restart (with resampling) is more appropriate at an I/O device such as a moving head disk where the data transfer time is typically small compared to seek and latency (justifying the restart assumption) and service time depends on the physical location of the last interrupting request's data (justifying the resampling assumption). To answer the question regarding improvement in CPU utilization based on the results of Section III, we plot CPU utilization as a function of the batch CPU service time $\gamma$ for various $N$, $M$. Figure 4a gives the results for $(N, M) = (2, 0), (2, 2), (2, 10), (2, \infty)$ and $(5, 0), (5, 2), (5, 10), (5, \infty)$. When the high-priority multiprogramming level $N = 2$, we see that the batch CPU times must be of the order of 10–20 ms before significant improvement in CPU utilization occurs, and for times in excess of 40 ms, almost complete CPU utilization is attained. For the case $N = 5$, the batch CPU times need to be 100–200 ms to get significant improvement, with 320 ms being the time for almost complete CPU utilization. These results can be anticipated using the rule of thumb developed in Remark ($v$) of Section III. For this example, the high-priority traffic experiences a

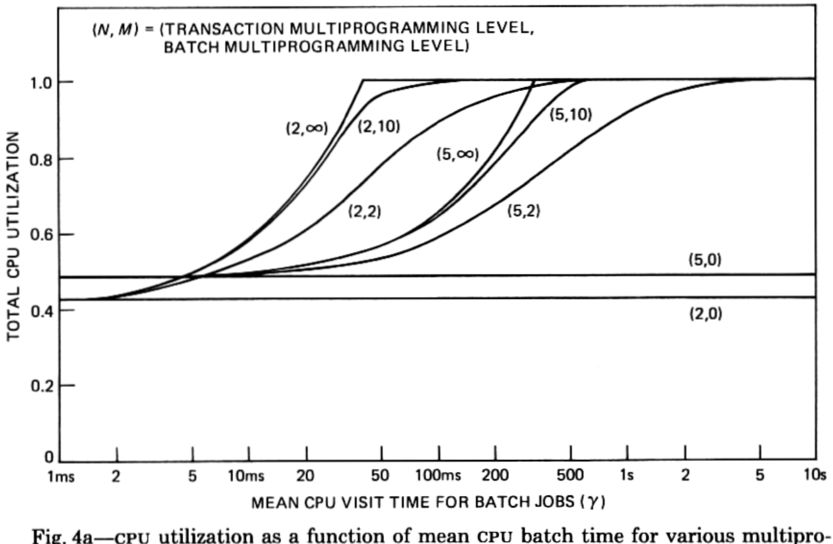


Fig. 4a—CPU utilization as a function of mean CPU batch time for various multiprogramming levels when transactions receive priority at both the CPU and I/O device.

bottleneck of strength 2. If this bottleneck is stronger, or the high-priority multiprogramming level is larger, our rule of thumb shows that the batch work has to be much more strongly CPU-bound to justify its introduction.

In the above arrangement, the batch work needs to be heavily CPU-bound to make its introduction worthwhile because batch I/O requests encounter the transaction bottleneck at the I/O device. Even though a batch job may only need infrequent I/O, it is often prevented from continuing by transaction I/O. In such circumstances, one might consider giving batch jobs high priority for I/O since, with appropriate parameters, batch jobs will rarely hold up transactions. The performance of such an arrangement is now evaluated using the results for model B. Figure 4b gives the results for the same parameters as before but with $(N, M) = (2, 0), (2, 2), (2, \infty)$ and $(5, 0), (5, 2)$; for this arrangement, we must show high-priority CPU utilization as well as total CPU utilization, since the former varies with $M$ and $\gamma$. We observe that the introduction of batch jobs at a low multiprogramming level can yield a considerable increase in total CPU utilization. This increase can be accomplished with only a small effect on transaction throughput provided $\gamma$ (the batch CPU service time) is 50 to 100 ms or larger. For $\gamma$ comparable or smaller than the transaction CPU service time of 5 ms, a large degradation of transaction throughput occurs. If $\gamma$ is smaller than 10 ms, then as $M$ increases, transaction throughput approaches zero. For certain parameter combinations (e.g., $N = 5$, $\gamma = 100$ ms) the latter arrangement offers a larger improvement in total CPU utilization

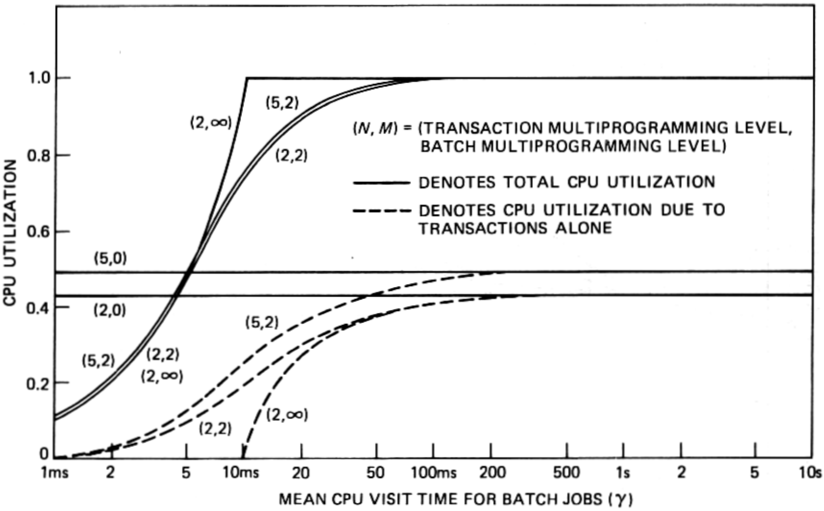


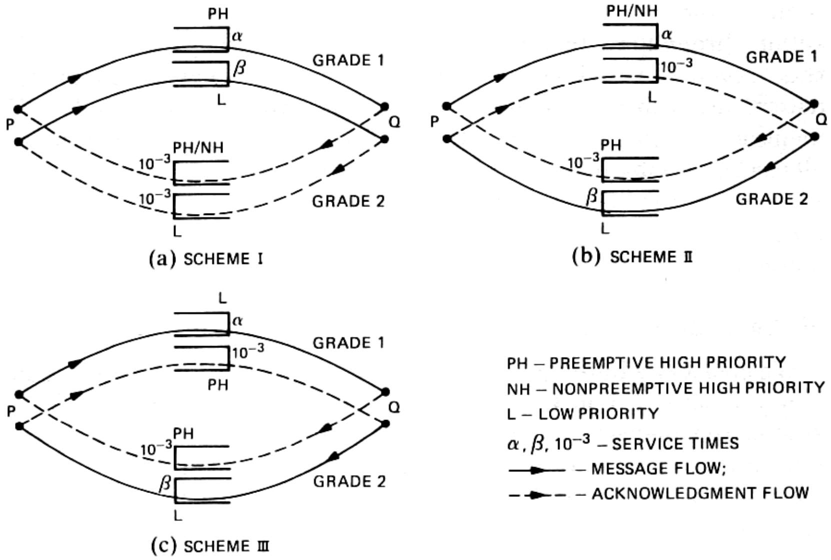
Fig. 4b—CPU utilization as a function of mean CPU batch time for various multiprogramming levels when transactions receive CPU priority and batch jobs receive I/O priority.

than the former, with only minor accompanying degradation of transaction throughput. In general, the extra total throughput offered by this "reversed priorities" arrangement must be weighed against any deterioration in transaction service, as quantified by the model. We mention that there may be a hazard in such a priority scheme since if all the batch jobs are simultaneously undergoing an abnormal flurry of I/O activity (or have actual mean service times substantially smaller than those being modeled) transaction processing might be temporarily halted as discussed in Remark (*iii*) of Section IV.

## VI. DATA COMMUNICATIONS EXAMPLE

We consider a full-duplex communication channel terminated at end points labeled $P$ and $Q$ by front-end communication processors. We assume the following simple transmission protocol. Messages are transmitted from one endpoint to the other and individual short acknowledgments are returned in the reverse direction. The acknowledgments serve the dual purpose of error control and flow control, and an endpoint must stop transmitting when it has a number $W$ of outstanding acknowledgments. Such a flow control scheme is often referred to as window flow control, and $W$ the window length. This protocol has been studied by Reiser in the network context using closed queuing networks with suitable heuristics to approximate the effect of different message sizes at first-come first-served queues and prioritized acknowledgments.[5] Our models here are less sophisticated with a two-node queuing network representing a single full-duplex channel, but they do allow some exact results for two chains with different message sizes and priority. The questions we seek to answer relate primarily to the effect of providing two grades of service between points $P$ and $Q$.

We assume that the end points $P$ and $Q$ return an acknowledgment as soon as they complete receiving a message. This is tantamount to assuming that the front-end processors are fast in comparison to the data links and have sufficient memory space so that acknowledgments are rarely withheld for purposes of flow control. We are also assuming that the data channels are essentially error free and retransmissions are rarely needed. Messages and acknowledgments are assumed to require an exponentially distributed time for transmission through the data channel. This assumption is more reasonable for messages than for acknowledgments where it is tolerable since acknowledgments are usually relatively short. There are two grades of service available, referred to as grades 1 and 2. Grade 1 service is regarded as having premium throughput characteristics, whereas grade 2 is designed to operate in a background mode to obtain increased use of the channel. We consider three configurations, referred to as schemes I–III, which are shown in Fig. 5.


(a) SCHEME I

(b) SCHEME II

(c) SCHEME III

Fig. 5—Three schemes to prioritize a data link.

### 6.1 Three schemes to prioritize a data link

#### Scheme I

We consider a grade 1 transfer to be in progress from $P$ to $Q$ and evaluate the possibility of introducing grade 2 message flow in the same direction. In order that grade 1 service be minimally affected by the introduction of grade 2 service, we specify that grade 1 messages and acknowledgments receive higher transmission priority than grade 2 messages and acknowledgments. This configuration is shown in Fig. 5a. Preemption of message is reasonable in packetized or framed transmission where data streams are, or can be, broken into smaller parts for transmission. Depending on the implementation, preemption of acknowledgments may or may not be possible and we consider both possibilities. Hence, we use model A where we identify the left (respectively right) node with the $Q$-to-$P$ (respectively $P$-to-$Q$) channel of the full-duplex link. Suppose an acknowledgment takes a mean of 1 ms for transmission, a grade 1 message an average of $\alpha$ seconds, and a grade 2 message an average of $\beta$ seconds. Let the window size for grade 1 (respectively grade 2) messages be $N$ (respectively $M$). Then taking, for example, $N = M = 4$, $\alpha = \beta = 3$ ms and allowing preemption of grade 2 acknowledgments, we find that grade 1 throughput is 330.58 messages/s and grade 2 throughput is 2.72 messages/s, i.e., grade 2 service accounts for only 0.8 percent of the total utilization of the $P$-to-$Q$ channel. If we disallow preemption of acknowledgments, then the model described in Remark (*vii*) of Section III is applicable and yields 330.56 messages/s and 2.73 messages/s as approximations to the grade 1 and 2 throughputs, respectively. The criterion for validity of the approximation reads $1/1024 \ll 1$ in this case. These results are easily anticipated since grade 1 messages already utilize 99.2 percent of the $P$-to-$Q$ channel and there is little point in introducing grade 2 service in the same direction.

### *Scheme II*

When there is grade 1 message transfer from $P$ to $Q$ and only acknowledgment traffic from $Q$ to $P$, it would seem worthwhile to carry grade 2 messages from $Q$ to $P$ assuming, of course, that there is a demand. As before, grade 2 messages receive lower preemptive priority and grade 2 acknowledgments lower preemptive or nonpreemptive priority. This arrangement, shown in Figure 5b, calls for use of model A, but this time, with the left (respectively right) node identified with the $P$-to-$Q$ (respectively $Q$-to-$P$) channel. We take the same definitions for $N$, $M$, $\alpha$, $\beta$ and a 1-ms acknowledgment time. Then with $N = M = 4$, $\alpha = \beta = 3$ ms and preemption of acknowledgments, we find that grade 1 throughput is 330.58 messages/s and grade 2 throughput is 7.14 messages/s. Without preemption of acknowledgments, throughputs become 328.99 and 11.87, respectively, and the criterion for validity of the approximation reads $1/64 \ll 1$. When $\beta$ is increased to 60 ms, grade 2 throughput (with preemption of acknowledgments) becomes 5.63 messages/s but the messages are 20 times longer so total effective message data throughput is increased by 34 percent. Without preemption of acknowledgments, throughputs become 7.79 messages/s for grade 2 and 329.54 messages/s for grade 1. This is a 47 percent increase in data throughput. In this case, the criterion for validity of the approximation is $1/976 \ll 1$. As expected, grade 2 service in the opposing direction only attains a significant throughput when its messages are much longer than those of grade 1. When this is not the case, the grade 1 messages cause an impediment to grade 2 acknowledgments that prevents a worthwhile grade 2 throughput.

### *Scheme III*

The priority given to both grade 1 messages and acknowledgments in Scheme II reflects a reluctance to allow grade 1 service to be more than minimally degraded by grade 2 service. The $Q$-to-$P$ channel is underutilized because the grade 2 acknowledgments suffer the grade 1 bottleneck in the $P$-to-$Q$ channel. But since acknowledgments are relatively short, we now ask how much grade 2 service improves and grade 1 service deteriorates when all acknowledgments receive priority over all messages. This arrangement is shown in Fig. 5c and $N$, $M$, $\alpha$,

$\beta$ are defined as previously. By using the results of model B, a 1-ms acknowledgment time and $\alpha = \beta = 3$ ms, grade 1 and 2 throughputs are found to be 248 messages/s and both channels are 99.4 percent utilized. Introducing grade 2 traffic has yielded a 50-percent increase in total traffic carried, but at the expense of a 25-percent degradation in grade 1 throughput. The effect of grade 2 acknowledgments on grade 1 messages is reduced when $\beta$ is increased. For example, when $\beta$ is 6 ms, grade 1 throughput is 292 messages/s and grade 2 throughput is 118 messages/s. Now grade 2 service has yielded a 60-percent increase in total message data throughput at the expense of a grade 1 throughput degradation of 12 percent. The introduction of grade 2 service has caused grade 1 message delay to increase from 10.65 ms to 12.30 ms and grade 1 acknowledgment delay remains at 1.45 ms (see Remark (*iv*) of Section IV). Utilization of the $P$-to-$Q$ (respectively $Q$ to $P$) channel is now 99.3 percent (respectively 99.95 percent).

As already stated in Section IV, a system with priorities allocated in such a way will tend to alternate between periods where customers of only one type (grade in this case) are processed. Hence, for this scheme, we need to make certain that during periods when grade 2 service is occurring grade 1 performance is not significantly disrupted in the short term. In this case, the fact that the 1-ms acknowledgment time is considerably shorter than $\beta$, makes it unlikely that a second grade 2 message will complete transmission before the acknowledgment from the former message is returned and, hence, that grade 1 service will be able to continue without an intolerably long delay. On the other hand, if the grade 2 source were to send a long sequence of very short messages, grade 1 communications could be interrupted for a considerable period. In an actual implementation, it might be desirable to incorporate a mechanism to prevent this occurrence.

Although we have only examined a limited set of parameter values, we can summarize the results of this section. In the absence of rather extreme traffic parameters, there is little justification for introduction of a lower grade of service which operates essentially in a background mode. On the other hand, if some degradation of the premium service grade is tolerable, then otherwise unused channel capacity can carry an appreciable amount of lower-grade traffic.

## VII. COMPARISON WITH AN APPROXIMATION TECHNIQUE

Our final application will be an evaluation of a convenient and commonly used approximation technique for handling priorities.[3,4,5] The technique considers the low-priority customers at a node to have a dedicated server of rate $\mu_L\ (1 - \rho_H)$, where $\mu_L$ is the low-priority service rate and $\rho_H$ is the utilization due to high-priority customers at that node. As noted in Ref. 5, this approximation is justifiable when the interruptions caused by high-priority traffic are frequent but of short duration. This suggests a criterion (sufficient condition) for satisfactory accuracy of the approximation: the high-priority busy cycle length at a node should be short in comparison with the low-priority service time at the same node.

Table I shows some results for model A with various parameter combinations $N$, $M$, $\nu$, $\mu$, $\lambda$. We tabulate the exact throughput $T$ and mean delay $D$ at each node for the low-priority customers using the results of Section II, and the approximations to these quantities based on the above approximation technique. We also tabulate the high-priority mean busy cycle B at each node to enable comparison of approximation accuracy with degree of satisfaction of the above criterion. For $T$, $D$, $B$ the subscripts $H$, $L$ distinguish high and low priority and $l$, $r$ distinguish left and right nodes. Note that $B_{H,l} = (\nu - \nu^{-N})/(\nu - 1)$, $B_{H,r} = (\nu^{-1} - \nu^{N})/(1 - \nu)$.

In Table I note that when the criterion is satisfied at both nodes (cases 1, 2, 6, 7), the approximation is quite successful with errors of less than 2 percent. When the criterion is violated at one node only (cases 3, 8, 9), the approximate results might be regarded as satisfactory or unsatisfactory, depending on one's viewpoint. When the criterion is violated at both nodes (cases 4, 5, 10), both approximate throughputs and delays show large errors. Case 5 reflects a rather extreme choice of parameters and is included only to show the large errors which are theoretically possible.

Another vehicle for examining the effectiveness of the approximation technique is to compare it with the exact results for a homogeneous open network as considered in Section II. The approximation yields an expression for class $i$ delay (including service time) at node $j$ of

$$D_j^i \approx \mu_j^{-1} \bigg/ \left(1 - \sum_{k=i}^{P} \rho_j^k\right),$$

which, in comparison with the exact result, is seen to be too small by a factor of

$$1 - \sum_{k=i+1}^{P} \rho_j^k .$$

Hence, for this type of network we would anticipate significant error if the approximation were applied to a priority class when higher-priority classes utilize a significant portion of a node's processing capacity. Indeed, the homogeneous network is a challenging test of the approximation technique since interruptions of a customer's service are of a duration at least comparable with the service time; our earlier criterion is never satisfied for such a network.

Table I—Comparison of the exact results for model A with an approximation technique

| Case | $N$ | $M$ | $\nu$ | $\mu$ | $\lambda$ | $B_{H,l}$ | $B_{H,r}$ | Technique | $T_L$ | $D_{L,l}$ | $D_{L,r}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 10 | 1.0 | 0.1 | 0.1 | 2 | 2.00 | exact | 0.0450 | 111 | 111 |
| | | | | | | | | approx. | 0.0455 | 110 | 110 |
| 2 | 3 | 5 | 1 | 0.03 | 0.03 | 4 | 4.00 | exact | 0.00616 | 406 | 406 |
| | | | | | | | | approx. | 0.00625 | 400 | 400 |
| 3 | 3 | 5 | 1 | 0.03 | 1.00 | 4 | 4.00 | exact | 0.750(−2) | 656 | 10.5 |
| | | | | | | | | approx. | 0.750(−2) | 663 | 4.12 |
| 4 | 3 | 5 | 1 | 1.00 | 1.00 | 4 | 4.00 | exact | 0.139 | 18.0 | 18.0 |
| | | | | | | | | approx. | 0.208 | 12.0 | 12.0 |
| 5 | 3 | 5 | 1 | 100.00 | 100.00 | 4 | 4.00 | exact | 0.416 | 6.12 | 6.12 |
| | | | | | | | | approx. | 20.8 | 0.12 | 0.12 |
| 6 | 1 | 10 | 0.2 | 0.02 | 0.02 | 6 | 6.00 | exact | 0.00333 | 2923 | 77.5 |
| | | | | | | | | approx. | 0.00333 | 2925 | 75.0 |
| 7 | 3 | 5 | 0.2 | 0.0005 | 0.01 | 156 | 6.24 | exact | 0.321(−5) | 0.156(7) | 128 |
| | | | | | | | | approx. | 0.321(−5) | 0.156(7) | 125 |
| 8 | 3 | 5 | 0.2 | 0.1 | 0.01 | 156 | 6.24 | exact | 0.641(−3) | 7646 | 156 |
| | | | | | | | | approx. | 0.641(−3) | 7664 | 136 |
| 9 | 3 | 5 | 0.2 | 0.0005 | 10.00 | 156 | 6.24 | exact | 0.321(−5) | 0.156(7) | 3.56 |
| | | | | | | | | approx. | 0.321(−5) | 0.156(7) | 0.125 |
| 10 | 3 | 5 | 0.2 | 1.00 | 10.00 | 156 | 6.24 | exact | 0.00608 | 819 | 3.69 |
| | | | | | | | | approx. | 0.00641 | 780 | 0.125 |

Note: $(x)$ indicates $\times 10^{x}$

## VIII. CONCLUSIONS

We have seen that the analysis of queuing networks is somewhat involved when local balance is not satisfied but that some useful results can still be obtained. It is clear that further results are needed to extend the applicability of these models. Section VII shows that further attention should also be directed towards establishing and improving the range of validity of existing approximation techniques.

## REFERENCES

1. F. Baskett et al., "Open, Closed, and Mixed Networks of Queues with Different Classes of Customers," J. of ACM, *22,* No. 2 (April 1975), pp. 248–60.
2. F. P. Kelly, *Reversibility and Stochastic Networks*, New York: John Wiley, 1979.
3. M. Reiser, "Interactive Modeling of Computer Systems," IBM Systems Journal, No. 4 (1976), pp. 309–27.
4. K. C. Sevcik, "Priority Scheduling Disciplines in Queuing Network Models of Computer Systems," *Information Processing 77,* B. Gilchrist, Ed., IFIP, Amsterdam: North Holland Publishing Co., 1977.
5. M. Reiser, "A Queuing Network Analysis of Computer Communication Networks with Window Flow Control," IEEE Trans. on Comm., *COM-27,* No. 8 (August 1979), pp. 1199–1209.
6. N. K. Jaiswal, *Priority Queues*, New York: Academic Press, 1968.
7. B. Avi-Itzhak and D. P. Heyman, "Approximate Queuing Models for Multiprogramming Computer Systems," Operations Research, *21,* No. 6 (1973), pp. 1212–30.
8. W. J. Gordon and G. F. Newell, "Closed Queuing Systems with Exponential Servers," Operations Research, *15*(1967), pp. 254–65.
9. M. Reiser, "Mean Value Analysis of Queuing Networks, A New Look at an Old Problem," *Performance of Computer Systems*, M. Arato, A. Butrimenko, E. Gelenbe (Eds.), Amsterdam: North-Holland Publishing Company, 1979, pp. 63–77.
10. J. R. Jackson, "Jobshop-like Queuing Systems," Management Science, *10,* No. 1 (October 1963), pp. 131–42.
11. H. White and L. S. Christie, "Queuing with Preemptive Priorities or Breakdown," Operations Research, *6,* No. 1 (January 1958), pp. 79–95.
12. L. Kleinrock, *Communication Nets*, New York: McGraw-Hill, 1964.
13. L. Kleinrock, *Queueing Systems*, Vol. II, New York: John Wiley, 1976.
14. U. Herzog, L. Woo, and K. M. Chandy, "Solution of Queuing Problems by a Recursive Technique," IBM J. Res. Develop., (May 1975), pp. 295–300.