

DEDUCING QUEUEING FROM TRANSACTIONAL DATA: THE QUEUE INFERENCE ENGINE, REVISITED

DIMITRIS J. BERTSIMAS

Massachusetts Institute of Technology, Cambridge, Massachusetts

L. D. SERVI

GTE Laboratories, Incorporated, Waltham, Massachusetts

(Received April 1990; revision received February 1991; accepted July 1991)

R. Larson proposed a method to statistically infer the expected transient queue length during a busy period in $O(n^5)$ solely from the n starting and stopping times of each customer's service during the busy period and assuming the arrival distribution is Poisson. We develop a new $O(n^3)$ algorithm which uses these data to deduce transient queue lengths as well as the waiting times of each customer in the busy period. We also develop an $O(n)$ on-line algorithm to dynamically update the current estimates for queue lengths after each departure. Moreover, we generalize our algorithms for the case of a time-varying Poisson process and also for the case of i.i.d. interarrival times with an arbitrary distribution. We report computational results that exhibit the speed and accuracy of our algorithms.

In 1989, Larson proposed the fascinating problem of inferring the transient queue length behavior of a system solely from data about the starting and stopping times of each customer service. He noted that automatic teller machines (ATMs) at banks have these data, yet they are unable to directly observe the queue lengths. Also, in a mobile radio system one can monitor the airwaves and again obtain times of call setups and call terminations although one is unable to directly measure the number of potential mobile radio users who are awaiting a channel. Other examples include checkout counters at supermarkets, traffic lights, pay telephones, and nodes in a telecommunications network. In fact, the algorithm in Section 1 has been applied to transmission data from an ethernet line (Gawlick 1990). More generally, the problem arises when costs or feasibility prevents the direct observation of the queue, although measurements of the departing epochs are possible. This paper discusses the deduction of the queue behavior only from these transactional data. In this analysis, no assumptions are required regarding the service time distribution: it could be state-dependent, dependent on the time of day, with or without correlations. Moreover, the service discipline is arbitrary. The only assumption is that all servers are busy when there are customers in the queue. Furthermore, the number of servers can be arbitrary.

In Larson (1990) two algorithms are proposed to

solve this problem based on two astute observations: 1) the beginning and ending of each busy period can be identified by a service completion time which is not immediately followed by a new service commencement; 2) a service commencement at time t_i implies that the arrival time of the corresponding customer must have arrived between the beginning of the arrival time of the previous customer and t_i . Furthermore, if the arrival process is known to be Poisson, then the a posteriori probability distribution of the arrival time of this customer must be uniformly distributed in this interval. Larson (1990) uses these observations to derive an $O(n^5)$ and an $O(2^n)$ algorithm to compute the transient queue lengths during a busy period of n customers.

In this paper, we propose an $O(n^3)$ algorithm, which estimates the transient queue length during a busy period as well as the delay of each customer served during the busy period. Moreover, we generalize the algorithm for the case of a time-varying Poisson process to another $O(n^3)$ algorithm and also find an algorithm for the case of stationary interarrival times from an arbitrary distribution. We also develop an $O(n)$ on-line algorithm to dynamically update the current estimates for queue lengths after each departure. This algorithm is similar to Kalman filtering in structure in that the current estimates for future queue lengths are updated dynamically in real time.

The paper is structured as follows: In Section 1, we

Subject classifications Queues: statistical inference, algorithms, transient results.
Area of review STOCHASTIC PROCESSES AND THEIR APPLICATIONS.

describe the exact $O(n^3)$ algorithm and prove its correctness assuming that the arrival process is Poisson. In Section 2, we generalize the algorithm for the case of a time-varying Poisson process into a new $O(n^3)$ algorithm. In Section 3, we further generalize our methods to handle the case of an arbitrary stationary interarrival time distribution. In Section 4, we describe the algorithm to dynamically update queue lengths in real time. In Section 5, we report numerical results. We also examine the sensitivity of the estimates to the arrival process.

1. STATIONARY POISSON ARRIVALS

In this section, we will assume that the arrival process to the system can accurately be modeled as a Poisson process with an arrival rate that it is unknown but constant within a busy period. The arrival rate, however, could vary from busy period to busy period. In Sections 2 and 3 this assumption is relaxed. We consider the following problem: Given only the departure epochs (service completions) $t_i, i = 1, 2, \dots, n$ during a busy period starting at time 0 and having exactly n customers, estimate the queue length distribution at time t and the waiting time of each customer. As noted, we make no assumptions on the service time distribution or the number of servers.

1.1. Notation and Preliminaries

The following definitions are needed. Let

- n = the number of customers served during the last busy period;
- t_i = the time of the i th service completion, $i = 1, 2, \dots, n$;
- X_i = the (unknown) arrival time of the i th customer, $i = 1, 2, \dots, n$;
- $N(t)$ = the cumulative number of arrivals in $[0, t]$;
- $Q(t)$ = the number of customers in the queue at time t^- ;
- $D(t)$ = the cumulative number of customers who departed from the queue and entered service in $[0, t]$ (so $D(t) = j$ for $t_{j-1} < t \leq t_j$);
- W_i = the waiting time of the i th customer;
- $(O(\vec{i}))$ = the event $\{X_2 \leq t_1, \dots, X_n \leq t_{n-1}\}$;
- $O'(\vec{i}, n)$ = the event $\{X_2 \leq t_1, \dots, X_n \leq t_{n-1}, \text{ and } N(t_n) = n\}$.

For the case of multiserver systems we define the busy period as beginning when all servers are occupied and ending when one of the servers first becomes idle. We

assume that the last busy period started at time 0, so $X_1 = 0$. Note also that $N(t)$ and $D(t)$ include the customer that initiated the busy period. Therefore, $\Pr\{N(t) = k\} = e^{-\lambda t}(\lambda t)^{k-1}/(k-1)!, k = 1, \dots$.

We observe the process $D(t)$ and wish to characterize the distribution of $N(t)$ given $D(t)$. From $N(t)$, properties of $Q(t)$ and W_j can be computed directly. For example, $Q(t) = N(t) - D(t)$. Suppose that exactly $n - 1$ consecutive service completions coincide with new service initiations at time t_1, t_2, \dots, t_{n-1} , and a service completion takes place at time t_n without a simultaneous new service initiation. Then, one knows that event $O'(\vec{i}, n)$ occurred, i.e., $X_2 \leq t_1, \dots, X_n \leq t_{n-1}$ and $N(t_n) = n$. We want to compute the distribution of $N(t)$ conditioned on $O'(\vec{i}, n)$.

To do this note (Ross 1983) that the conditional joint density for $0 \leq x_1 \leq x_2 \dots \leq x_n$

$$f(X_2 = x_1, \dots, X_n = x_{n-1} | N(x_n) = n) = \frac{(n-1)!}{x_n^{n-1}} \tag{1}$$

As a result

$$\Pr\{X_2 \leq t_1, \dots, X_n \leq t_{n-1} | N(t_n) = n\} = \frac{(n-1)!}{t_n^{n-1}} \int_{x_2=0}^{t_1} \int_{x_3=x_2}^{t_2} \dots \int_{x_n=x_{n-1}}^{t_{n-1}} dx_2 \dots dx_n. \tag{2}$$

Equation 2 is an example of an integral that encapsulates the a posteriori information of the arrival times. This information combined with the known departure times leads to a posteriori estimates of the queue lengths and waiting times. More generally, as will be shown in Section 1.2 and proven in Section 1.3 to evaluate this type of integral efficiently we must evaluate the following related integrals:

$$H_{j,k}(y) = \int_{x_2=0}^{t_1} \int_{x_3=x_2}^{t_2} \dots \int_{x_j=x_{j-1}}^{t_{j-1}} \int_{x_{j+1}=x_j}^y \dots \int_{x_k=x_{k-1}}^y dx_2 \dots dx_k \tag{3}$$

and

$$F_k(y) = \int_{x_{k+1}=0}^y \int_{x_{k+2}=x_{k+1}}^{t_{k+1}} \dots \int_{x_n=x_{n-1}}^{t_{n-1}} dx_{k+1} \dots dx_n \tag{4}$$

at $y = t_j$.

Define

$$\begin{aligned}
 h_{j,k} &= H_{j,k}(t_j) \\
 &= \int_{\lambda_2=0}^{t_1} \int_{\lambda_3=\lambda_2}^{t_2} \cdots \int_{\lambda_j=\lambda_{j-1}}^{t_{j-1}} \int_{\lambda_{j+1}=\lambda_j}^{t_j} \\
 &\quad \cdots \int_{\lambda_k=\lambda_{k-1}}^{t_j} dx_2 \cdots dx_k
 \end{aligned} \quad (5)$$

and

$$\begin{aligned}
 f_{j,k} = F_k(t_j) &= \int_{\lambda_{k+1}=0}^{t_j} \int_{\lambda_{k+2}=\lambda_{k+1}}^{t_{k+1}} \\
 &\quad \cdots \int_{\lambda_n=\lambda_{n-1}}^{t_{n-1}} dx_{k+1} \cdots dx_n.
 \end{aligned} \quad (6)$$

The first four steps of the following algorithm will compute these values in $O(n^3)$ and the next two steps will use these values to compute transient queue length and waiting time estimates. The validity of the algorithm will be demonstrated in Section 1.3.

1.2. An Exact $O(n^3)$ Algorithm

Step 0. (Initialization). Let

$$h_{1,1} = 1, f_{j,n} = \delta_{j,n}. \quad (7)$$

Step 1. (Calculation of the diagonal elements $h_{k,k}$).

For $k = 2$ to n ,

$$h_{k,k} = \sum_{i=1}^{k-1} (-1)^{k-i+1} \frac{t_i^{k-i}}{(k-i)!} h_{i,i}. \quad (8)$$

Step 2. (Calculation of $h_{j,k}$). For $j = 1$ to $n-1$, for $k = j+1$ to n ,

$$h_{j,k} = \frac{t_j^{k-1}}{(k-1)!} - \sum_{i=1}^{j-1} \frac{(t_j - t_i)^{k-i}}{(k-i)!} h_{i,i}. \quad (9)$$

Step 3. (Calculation of the diagonal elements $f_{k,k}$).

For $k = n-1$ to 1 ,

$$f_{k,k} = \sum_{i=k}^{n-1} (-1)^{i-k} \frac{t_i^{i-k+1}}{(i-k+1)!} f_{i+1,i+1}. \quad (10)$$

Step 4. (Calculation of $f_{j,k}$). For $j = 1$ to $n-2$, for $k = j+1$ to $n-1$,

$$f_{j,k} = \sum_{i=k}^{n-1} (-1)^{i-k} \frac{t_j^{i-k+1}}{(i-k+1)!} f_{i+1,i+1}. \quad (11)$$

Step 5. (Estimates of transient queue length).

1. For $j = 1$ to $n-1$

$$E[Q(t_j) | O'(\vec{i}, n)] = \frac{\sum_{k=j+1}^n kh_{j,k}[f_{k,k} - f_{j,k}]}{h_{n,n}} - j. \quad (12)$$

2. Given $t, t_{j-1} < t \leq t_j$

$$\begin{aligned}
 E[Q(t) | O'(\vec{i}, n)] &= (1 - \theta)E[Q(t_{j-1}) | O'(\vec{i}, n)] \\
 &\quad + \theta E[Q(t_j) | O'(\vec{i}, n)]
 \end{aligned} \quad (13)$$

where $\theta = (t - t_{j-1}) / (t_j - t_{j-1})$ and for $0 \leq k \leq n - j$

$$\begin{aligned}
 \Pr\{Q(t) = k | O'(\vec{i}, n)\} \\
 &= \frac{H_{j,k+j}(t)[F_{k+j}(t_{k+j}) - F_{k+j}(t)]}{h_{n,n}},
 \end{aligned} \quad (14)$$

and

$$\Pr\{Q(t_n) = 0 | O'(\vec{i}, n)\} = 1 \quad (15)$$

where for $j = 2, 3, \dots, n$ and $k = j, j+1, \dots, n$,

$$H_{j,k}(y) = \frac{y^{k-1}}{(k-1)!} - \sum_{i=1}^{j-1} \frac{(y - t_i)^{k-i}}{(k-i)!} h_{i,i}, \quad (16)$$

and for $k = 1, 2, \dots, n-1$,

$$F_k(y) = \sum_{i=k}^{n-1} \frac{(-1)^{i-k} y^{i-k+1}}{(i-k+1)!} f_{i+1,i+1}. \quad (17)$$

where $h_{1,1}$ and $f_{n,n}$ are defined in (7).

Step 6. (Estimates of transient waiting time).

$$\begin{aligned}
 E[W_k^m] &\geq (t_k - t_1)^m - \sum_{j=2}^k [(t_k - t_{j-1})^m - (t_k - t_j)^m] \\
 &\quad \cdot \sum_{i=j}^{k-1} \left(\frac{h_{j-1,i}[f_{i,i} - f_{j-1,j}]}{h_{n,n}} \right).
 \end{aligned} \quad (18)$$

$$\begin{aligned}
 E[W_k^m] &\leq t_k^m - \sum_{j=1}^{k-1} [(t_k - t_{j-1})^m - (t_k - t_j)^m] \\
 &\quad \cdot \sum_{i=j+1}^{k-1} \left(\frac{h_{j,i}[f_{i,i} - f_{j,j}]}{h_{n,n}} \right).
 \end{aligned} \quad (19)$$

For $t_{j-1} < t_k - t \leq t_j$

$$\begin{aligned}
 \Pr\{W_k \leq t | O'(\vec{i}, n)\} \\
 &= \frac{\sum_{r=1}^{k-1} H_{j,r+j}(t_k - t)[F_{r+j}(t_{r+j}) - F_{r+j}(t_k - t)]}{h_{n,n}},
 \end{aligned} \quad (20)$$

where $H_{j,k}(Y)$ and $F_k(y)$ are defined in (16) and (17), respectively.

From the first four steps of the algorithm all the $h_{j,k}$'s and $f_{j,k}$'s can be found in $O(n^3)$. From (12) and (13), to compute $E[Q(t) | O'(\vec{i}, n)]$ for only one value of t can be done in $O(n^2)$ because not all $h_{j,k}$'s and $f_{j,k}$'s are required. However, from (13), to compute $E[Q(t) | O'(\vec{i}, n)]$ for all values of t requires $O(n^3)$ with the bottleneck steps being Steps 2 and 4. To speed the implementation of the algorithm, for $j = 1$,

$2, \dots, n, j!$ should be computed once and stored as a vector. Also, note that the $f_{j,k}$'s and $h_{j,k}$'s need not be stored but instead can be computed and then immediately used in (12) to reduce the storage requirements of the algorithm to $O(n)$.

1.3. Proof of Correctness

In this subsection, we prove that indeed the algorithm correctly computes the distributions of the queue length and waiting time. A basic ingredient of the analysis is the following lemma that simplifies the multidimensional integrals.

Lemma 1

$$\int_{x_2=z}^y \int_{x_3=x_2}^y \dots \int_{x_k=x_{k-1}}^y dx_2 \dots dx_k = \frac{(y-z)^{k-1}}{(k-1)!} \tag{21}$$

Proof. (By induction). For $k = 2$, (21) is trivial. Suppose that (21) were true for $r = k$. Then, from the induction hypothesis,

$$\int_{x_2=z}^y \int_{x_3=x_2}^y \dots \int_{x_{k+1}=x_k}^y dx_2 \dots dx_{k+1} = \int_{x_2=z}^y \frac{(y-x_2)^{k-1}}{(k-1)!} dx_2 = \frac{(y-z)^k}{k!}.$$

Hence, (21) is true for $r = k + 1$ and the lemma follows by induction.

Remark. The proof of Lemma 1 can be generalized to show that

$$\int_{x_2=z}^y \int_{x_3=x_2}^y \dots \int_{x_k=x_{k-1}}^y f(x_k) dx_2 \dots dx_k = \int_{x_2=z}^y \frac{(y-x_2)^{k-2}}{(k-2)!} f(x_2) dx_2. \tag{22}$$

In the following proposition we justify Steps 1 and 2 of the algorithm.

Proposition 1. *Steps 1 and 2 of the algorithm are implied by the definition of $h_{j,k}$. Also, $H_{j,k}(y)$ defined in (3) satisfy (16).*

Proof. If we apply Lemma 1 to (3) we obtain for $j = 3, 4, \dots, n$ and $k = j, j + 1, \dots, n$,

$$H_{j,k}(y) = \int_{x_2=0}^{t_1} \int_{x_3=x_2}^{t_2} \dots \int_{x_j=x_{j-1}}^{t_{j-1}} \frac{(y-x_j)^{k-j}}{(k-j)!} dx_2 \dots dx_j, \tag{23}$$

which, after performing the innermost integral, is equal to

$$\int_{x_2=0}^{t_1} \int_{x_3=x_2}^{t_2} \dots \int_{x_{j-1}=x_{j-2}}^{t_{j-2}} \left[\frac{(y-x_{j-1})^{k-j+1}}{(k-j+1)!} - \frac{(y-t_{j-1})^{k-j+1}}{(k-j+1)!} \right] dx_2 \dots dx_{j-1}.$$

Hence, from (3), (5), and (23) we obtain

$$H_{j,k}(y) = H_{j-1,k}(y) - \frac{(y-t_{j-1})^{k-j+1}}{(k-j+1)!} h_{j-1,j-1}.$$

Successive substitution of this equation gives,

$$H_{j,k}(y) = H_{2,k}(y) - \sum_{i=2}^{j-1} \frac{(y-t_i)^{k-i}}{(k-i)!} h_{i,i}.$$

But from (23)

$$H_{2,k}(y) = \int_{x_2=0}^{t_1} \frac{(y-x_2)^{k-2}}{(k-2)!} dx_2 = \frac{y^{k-1}}{(k-1)!} - \frac{(y-t_1)^{k-1}}{(k-1)!}$$

and thus (16) follows for $j \geq 2$. Also, from (3) and (7), $H_{1,k}(y) = y^{k-1}/(k-1)!$, so (16) follows for $j = 1$. Since $h_{j,k} = H_{j,k}(t_j)$, (9) is obtained. Moreover, from (5) and (23), $h_{k,k} = H_{k,k}(0)$ and thus we obtain (8).

We now turn our attention to the justification of Steps 3 and 4 of the algorithm.

Proposition 2. *Steps 3 and 4 of the algorithm are implied by the definition of $f_{j,k}$ in (6). Also, $F_k(y)$ defined in (4) satisfy (17).*

Proof. From (4), we note that $F_k(y)$ is a polynomial in y of degree $n - k$. Let $c_{i,k}$ be the coefficients in

$$F_k(y) = \sum_{i=k}^{n-1} c_{k,i-k+1} y^{i-k+1}.$$

Moreover, from (4) for $k = 1, 2, \dots, n - 1$, we obtain:

$$F_k(y) = \int_{x_{k+1}=0}^y [F_{k+1}(t_{k+1}) - F_{k+1}(x_{k+1})] dx_{k+1}.$$

By differentiating,

$$\frac{dF_k(y)}{dy} = F_{k+1}(t_{k+1}) - F_{k+1}(y) \tag{24}$$

and

$$\frac{d^2F_k(y)}{dy^2} = -\frac{dF_{k+1}(y)}{dy}.$$

More generally for $i \geq k$,

$$\frac{d^{i-k+1}F_k(y)}{dy^{i-k+1}} = (-1)^{i-k} \frac{dF_i(y)}{dy}.$$

From Taylor's Theorem,

$$c_{k,i-k+1} = \frac{d^{i-k+1}F_k(0)/dy^{i-k+1}}{(i-k+1)!}$$

and $c_{i,1} = dF_i(0)/dy$. Therefore, for $i \geq k$,

$$c_{k,i-k+1} = \frac{(-1)^{i-k}}{(i-k+1)!} c_{i,1}. \tag{25}$$

Evaluating (24) at $y = 0$ gives

$$c_{k,1} = \frac{dF_k(0)}{dy} = F_{k+1}(t_{k+1}),$$

since $F_{k+1}(0)$. As a result, for $k = 1, 2, \dots, n-1$,

$$c_{k-1,1} = F_k(t_k) = \sum_{i=k}^{n-1} c_{k,i-k+1} t_k^{i-k+1}.$$

From (25),

$$c_{k-1,1} = \sum_{i=k}^{n-1} \frac{(-1)^{i-k} t_k^{i-k+1}}{(i-k+1)!} c_{i,1},$$

where $c_{n-1,1} = 1$. Thus, $f_{k,k} = F_k(t_k) = c_{k-1,1}$ satisfy (10). In addition,

$$F_k(y) = \sum_{i=k}^{n-1} c_{k,i-k+1} y^{i-k+1},$$

which from (25) is

$$\sum_{i=k}^{n-1} \frac{(-1)^{i-k} y^{i-k+1}}{(i-k+1)!} c_{i,1}$$

and thus (17) follows. Moreover, since $f_{j,k} = F_k(t_j)$, (11) follows.

Having established the validity of the steps of the algorithm required to compute the $h_{j,k}$'s and $f_{j,k}$'s we proceed to prove the correctness of the performance measure parts of the algorithm, Steps 5 and 6.

Theorem 1. *The distribution of $Q(t)$ and its mean conditioned on the observations $O'(\vec{t}, n)$ is given by (12), (13), (14) and (15) in Step 5 of the algorithm.*

Proof. For $n \geq k \geq j$ and $t_{j-1} < t \leq t_j$ the event $\{N(t) = k, O(\vec{t})\}$ occurs if and only if $0 \leq X_2 \leq t_1, \dots, X_{j-1} \leq X_j \leq t_{j-1}, X_j \leq X_{j+1} \leq t, \dots, X_{k-1} \leq X_k \leq t, t \leq X_{k+1} \leq t_k, \dots, X_{n-1} \leq X_n \leq t_{n-1}$. Since the arrival process is Poisson, from (1), the conditional density

function is uniform and hence,

$$\begin{aligned} \Pr\{N(t) = k, O(\vec{t}) \mid N(t_n) = n\} \\ = \frac{(n-1)!}{t_n^{n-1}} \int_{\lambda=0}^{t_1} \dots \int_{\lambda_j=\lambda_{j-1}}^{t_{j-1}} \int_{\lambda_{j+1}=\lambda_j}^t \\ \dots \int_{\lambda_k=\lambda_{k-1}}^t \int_{\lambda_{k+1}=t}^{t_k} \int_{\lambda_n=\lambda_{n-1}}^{t_{n-1}} dx_2 \dots dx_n, \end{aligned}$$

which, from (3) and (4), is

$$\frac{(n-1)!}{t_n^{n-1}} H_{j,k}(t)[F_k(t_k) - F_k(t)].$$

Similarly, from (2) and (5),

$$\begin{aligned} \Pr\{O(\vec{t}) \mid N(t_n) = n\} &= \frac{(n-1)!}{t_n^{n-1}} \\ &\cdot \int_{\lambda_2=0}^{t_1} \int_{\lambda_3=\lambda_2}^{t_2} \dots \int_{\lambda_n=\lambda_{n-1}}^{t_{n-1}} dx_2 \dots dx_n \\ &= \frac{(n-1)!}{t_n^{n-1}} h_{n,n}. \end{aligned}$$

Therefore, for $n-1 \geq k \geq j$ and $t_{j-1} < t \leq t_j$

$$\begin{aligned} \Pr\{N(t) = k \mid O(\vec{t}), N(t_n) = n\} \\ = \frac{H_{j,k}(t)[F_k(t_k) - F_k(t)]}{h_{n,n}} \end{aligned} \tag{26}$$

and, for $t_{j-1} < t \leq t_j$,

$$\Pr\{N(t) = n \mid O(\vec{t}), N(t_n) = n\} = \frac{H_{j,n}(t)}{h_{n,n}}. \tag{27}$$

For $t = t_j$, using (5), (6), and (7), we have for $n \leq k \leq j$,

$$\Pr\{N(t_j) = k \mid O(\vec{t}), N(t_n) = n\} = \frac{h_{j,k}[f_{k,k} - f_{j,k}]}{h_{n,n}}. \tag{28}$$

Since $Q(t) = N(t) - D(t)$ and $D(t_j) = j$

$$\begin{aligned} E[Q(t_j) \mid O(\vec{t}), N(t_n) = n] \\ = E[N(t_j) \mid O(\vec{t}), N(t_n) = n] - j \\ = \sum_{k=j+1}^n k \Pr\{N(t_j) = k \mid O(\vec{t}), N(t_n) = n\} - j \end{aligned}$$

because, for $k \leq j$, $\Pr\{N(t_j) = k \mid O(\vec{t}), N(t_n) = n\} = 0$. Hence, (12) follows from (28). Also, for $t_{j-1} < t \leq t_j$,

$$\begin{aligned} \Pr\{Q(t) = k \mid O(\vec{t}), N(t_n) = n\} \\ = \Pr\{N(t) = j + k \mid O(\vec{t}), N(t_n) = n\} \end{aligned}$$

so (14) follows from (26). In addition, (15) follows from the observation that $N(t_n) = D(t_n) = n$ with

probability 1 and $Q(t) = N(t) - D(t)$. Suppose that we condition on the event $\{N(t_{j-1}) = r\}$ and $\{N(t_j) = m\}$. Then for $t_{j-1} < t \leq t_j$,

$$\begin{aligned} & \Pr\{N(t) = k \mid O(\vec{t}), N(t_n) = n, N(t_{j-1}) = r, N(t_j) = m\} \\ &= [\Pr\{N(t) = k, N(t_{j-1}) = r, N(t_j) = m, \\ & \quad O(\vec{t}) \mid N(t_n) = n\}] / \\ & \quad [\Pr\{N(t_{j-1}) = r, (t_j) = m, O(\vec{t}) \mid N(t_n) = n\}] \\ &= \left[\int_{\lambda_2=0}^{t_1} \cdot \int_{\lambda_j=\lambda_{j-1}}^{t_{j-1}} \cdot \int_{\lambda_r=\lambda_{r-1}}^{t_{j-1}} \int_{\lambda_{r+1}=t_{j-1}}^t \cdot \int_{\lambda_k=\lambda_{k-1}}^t \int_{\lambda_{k+1}=t}^{t_j} \right. \\ & \quad \cdot \int_{\lambda_m=\lambda_{m-1}}^{t_j} \int_{\lambda_{m+1}=t_j}^{t_{m+1}} \cdot \int_{\lambda_n=\lambda_{n-1}}^{t_{n-1}} dx_2 \cdot dx_n \Big] / \\ & \quad \left[\int_{\lambda_2=0}^{t_1} \cdot \int_{\lambda_j=\lambda_{j-1}}^{t_{j-1}} \cdot \int_{\lambda_r=\lambda_{r-1}}^{t_{j-1}} \int_{\lambda_{r+1}=t_{j-1}}^{t_j} \right. \\ & \quad \cdot \int_{\lambda_m=\lambda_{m-1}}^{t_j} \int_{\lambda_{m+1}=t_j}^{t_{m+1}} \cdot \int_{\lambda_n=\lambda_{n-1}}^{t_{n-1}} dx_2 \cdot dx_n \Big] \\ &= \left[\int_{\lambda_{r+1}=t_{j-1}}^t \cdots \int_{\lambda_k=\lambda_{k-1}}^t \int_{\lambda_{k+1}=t}^{t_j} \right. \\ & \quad \left. \cdots \int_{\lambda_m=\lambda_{m-1}}^{t_j} dx_{r+1} \cdots dx_m \right] / \\ & \quad \left[\int_{\lambda_{r+1}=t_{j-1}}^{t_j} \cdots \int_{\lambda_m=\lambda_{m-1}}^{t_j} dx_{r+1} \cdots dx_m \right] \end{aligned}$$

after simplification. Using Lemma 1 and some additional simplification we obtain

$$\begin{aligned} & \Pr\{N(t) = k \mid O(\vec{t}), N(t_n) = n, N(t_{j-1}) = r, N(t_j) = m\} \\ &= \binom{m-r}{k-r} \theta^{k-1} (1-\theta)^{m-k} \end{aligned}$$

where $\theta = (t - t_{j-1}) / (t_j - t_{j-1})$. Hence,

$$\begin{aligned} & \Pr\{N(t) - N(t_{j-1}) = k' \mid O(\vec{t}), N(t_n) = n, N(t_{j-1}), N(t_j)\} \\ &= \binom{N(t_j) - N(t_{j-1})}{k'} \theta^{k'} (1-\theta)^{N(t_j) - N(t_{j-1}) - k'}. \quad (29) \end{aligned}$$

Therefore, $N(t)$ conditioned on $N(t_{j-1})$ and $N(t_j)$ has a binomial distribution so

$$\begin{aligned} & E[N(t) \mid O(\vec{t}), N(t_n) = n, N(t_{j-1}), N(t_j)] \\ &= N(t_{j-1}) + \theta(N(t_j) - N(t_{j-1})), \end{aligned}$$

which by taking expectations leads to

$$\begin{aligned} & E[N(t) \mid O(\vec{t}), N(t_n) = n] \\ &= (1 - \theta)E[N(t_{j-1})] + \theta E[N(t_j)]. \end{aligned}$$

Since $Q(t) = N(t) - j$ for $t_{j-1} < t \leq t_j$, the piecewise linearity of (13) follows.

Remark. Larson (1990) also contains a proof of the piecewise linearity of $E[N(t) \mid O(\vec{t}), N(t_n) = n]$. Our proof more easily generalizes to the case of time-varying arrival rates.

Finally, we establish the correctness of Step 6 of the algorithm.

Theorem 2. *The distribution of $W(t)$ and its moments conditioned on the observations $O'(\vec{t}, n)$ is given by (18), (19), and (20) in Step 6 of the algorithm.*

Proof. The m th moment of the k th customer waiting time is

$$E[W_k^m] = - \int_{t=0}^{\infty} t^m \frac{d \Pr\{W_k > t\}}{dt} dt$$

which, from integration by parts, is

$$\int_{t=0}^{\infty} mt^{m-1} \Pr\{W_k > t\} dt.$$

But the k th customer arrived in $[0, t_k]$ and began service at time t_k so $\Pr\{W_k > t\} = 0$ for $t > t_k$. Hence,

$$\begin{aligned} E[W_k^m] &= \int_{t=0}^{t_k} mt^{m-1} (1 - \Pr\{W_k \leq t\}) dt \\ &= \int_{t=0}^{t_k} mt^{m-1} dt - \sum_{j=1}^k \int_{t=t_k-t_j}^{t_k-t_{j-1}} mt^{m-1} \\ & \quad \cdot \Pr\{W_k \leq t\} dt \end{aligned}$$

with $t_0 = 0$. Since $-\Pr\{W_k \leq t\}$ decreases as t increases,

$$\begin{aligned} E[W_k^m] &\geq \int_{t=0}^{t_k} mt^{m-1} dt \\ & \quad - \sum_{j=1}^k \Pr\{W_k \leq t_k - t_{j-1}\} \int_{t=t_k-t_j}^{t_k-t_{j-1}} mt^{m-1} dt. \end{aligned}$$

But $W_k \leq t_k$ with probability 1 so

$$\begin{aligned} E[W_k^m] &\geq \int_{t=0}^{t_k} mt^{m-1} dt - \int_{t=t_k-t_1}^{t_k} mt^{m-1} dt \\ & \quad - \sum_{j=2}^k \Pr\{W_k \leq t_k - t_{j-1}\} \int_{t=t_k-t_j}^{t_k-t_{j-1}} mt^{m-1} dt \end{aligned}$$

which simplifies to

$$\begin{aligned} E[W_k^m] &\geq \int_{t=0}^{t_k-t_1} mt^{m-1} dt \\ & \quad - \sum_{j=2}^k \Pr\{W_k \leq t_k - t_{j-1}\} \int_{t=t_k-t_j}^{t_k-t_{j-1}} mt^{m-1} dt. \end{aligned}$$

But the wait of the k th customer, W_k , is simply $t_k - X_k$. Therefore, the event $\{W_k \leq t_k - t_{j-1}\}$ is the same as the event $\{X_k \geq t_{j-1}\}$ which is equivalent to the event $\{N(t_{j-1}) < k\}$. Hence,

$$E[W_k^m] \geq (t_k - t_1)^m - \sum_{j=2}^k \Pr\{N(t_{j-1}) < k\} \int_{t=t_k-t_j}^{t_k-t_{j-1}} mt^{m-1} dt \quad (30)$$

so (18) follows from (28), (30), and the observation that $N(t_{j-1}) > j - 1$. Using similar reasoning,

$$E[W_k^m] \leq \int_{t=0}^{t_k} mt^{m-1} dt - \sum_{j=1}^k \Pr\{W_k \leq t_k - t_j\} \int_{t=t_k-t_j}^{t_k-t_{j-1}} mt^{m-1} dt$$

and hence, because $\Pr\{W_k \leq 0\} = 0$,

$$E[W_k^m] \leq t_k^m - \sum_{j=1}^{k-1} \Pr\{N(t_j) < k\} \cdot \int_{t=t_k-t_j}^{t_k-t_{j-1}} mt^{m-1} dt \quad (31)$$

which, from (28), leads to (10). Also, from (30), (31), and $Q(t) = N(t) - D(t)$, it follows that

$$E[W_k^m] \geq (t_k - t_1)^m - \sum_{j=2}^k \Pr\{Q(t_{j-1}) < k - j + 1\} \cdot \int_{t=t_k-t_j}^{t_k-t_{j-1}} mt^{m-1} dt \quad (32)$$

and

$$E[W_k^m] \leq t_k^m - \sum_{j=1}^{k-1} \Pr\{Q(t_j) < k - j\} \cdot \int_{t=t_k-t_j}^{t_k-t_{j-1}} mt^{m-1} dt. \quad (33)$$

Using similar reasoning, for $t_{j-1} < t_k - t \leq t_j$

$$\begin{aligned} \Pr\{W_k \leq t \mid O'(\vec{t}, n)\} &= \Pr\{N(t_k - t) < k \mid O'(\vec{t}, n)\} \\ &= \Pr\{Q(t_k - t) < k - j \mid O'(\vec{t}, n)\} \end{aligned} \quad (34)$$

so (20) follows from (14).

2. THE TIME-VARYING POISSON ARRIVAL PROCESS

In the previous section, we assumed that the arrival rates are Poisson with a hazard rate λ that is constant

during each busy period and found the inferred queue length and waiting time. In some applications, however, it is not realistic to assume that λ remains constant. We will show in this section that all the results of the previous section are immediately extendible to this case without adding any complexity to the resulting algorithm. In particular, if the arrival rate is time-varying, $\lambda(t) > 0$, then we will show that *the algorithm in Section 1.2 is applicable if all t 's are replaced by $\int_0^t \lambda(x) dx$* . As before, we will be assuming that the busy period begins at time $X_1 = 0$.

Let $\lambda(t)$ be the time-varying arrival rate. We will first need to find the generalization of the conditional distribution of the order statistics given in (1).

Theorem 3. *Given that $N(x_n) = n$ and the time-varying arrival rate $\lambda(t)$, the conditional density function of the $n - 1$ arrival times X_2, X_3, \dots, X_n is*

$$f(X_2 = x_1, \dots, X_n = x_{n-1} \mid N(x_n) = n) = \frac{(n - 1)! \prod_{i=1}^{n-1} \lambda(x_i)}{\Lambda(x_n)^{n-1}}, \quad (35)$$

where $\Lambda(t) = \int_0^t \lambda(x) dx$.

Proof. The proof below is similar to Theorem 2.3.1 of Ross. Let $0 < x_1 < x_2 < \dots < x_n$, and let ϵ_i be small enough so that $x_i + \epsilon_i < x_{i+1}$, $i = 1, \dots, n - 1$. Now, $\Pr\{x_1 \leq X_2 \leq x_1 + \epsilon_1, \dots, x_{n-1} \leq X_n \leq x_{n-1} + \epsilon_n \text{ and } N(x_n) = n\} = \Pr\{\text{exactly one arrival in } [x_i, x_i + \epsilon_i] \text{ for } 1 \leq i \leq n - 1 \text{ and so arrival elsewhere in } (0, x_n]\}$.

But the number of Poisson arrivals in an interval I is Poisson with a mean $\int_{x \in I} \lambda(x) dx$, so the above probability is

$$\prod_{i=1}^{n-1} [e^{-\Lambda_i} M_i] e^{-\Lambda(x_n)} - \sum_{i=1}^{n-1} M_i = e^{-\Lambda(x_n)} \prod_{i=1}^{n-1} M_i$$

where

$$M_i = \int_{x_i}^{x_i + \epsilon_i} \lambda(x) dx = \lambda(x_i)\epsilon_i + o(\epsilon_i).$$

Since $\Pr\{N(x_n) = n\} = \Pr\{\text{exactly } n - 1 \text{ arrivals in } (0, x_n]\} = e^{-\Lambda(x_n)}(\Lambda(x_n))^{n-1}/(n - 1)!$ (35) follows.

Given that we have now characterized the joint conditional density we can answer questions about the system's behavior.

Proposition 3. *Suppose that $\lambda(t) > 0$. Then*

$$\Pr\{X_2 \leq t_1, \dots, X_n \leq t_{n-1} \mid N(t_n) = n\} = \frac{(n - 1)! \int_{\lambda_2=0}^{\Lambda(t_1)} \int_{\lambda_3=\lambda_2}^{\Lambda(t_2)} \dots \int_{\lambda_n=\lambda_{n-1}}^{\Lambda(t_{n-1})} dx_2 \dots dx_n}{\Lambda(t_n)^{n-1}}. \quad (36)$$

Proof. From (35),

$$\Pr\{X_2 \leq t_1, \dots, X_n \leq t_{n-1} \mid N(t_n) = n\} = \frac{(n-1)! \int_{\lambda_2=0}^{t_1} \int_{\lambda_3=\lambda_2}^{t_2} \dots \int_{\lambda_n=\lambda_{n-1}}^{t_{n-1}} \prod_{i=2}^n \lambda(x_i) dx_i}{\Lambda(t_n)^{n-1}}. \quad (37)$$

Consider now the transformation of variables $y_i = \Lambda(x_i)$, $i = 2, \dots, n$. This is a one-to-one monotonically increasing function if $\lambda(t) > 0$. Furthermore, $dy_i = \lambda(x_i) dx_i$. With this transformation of variables all of the upper limits change from $x_i = t_i$ to $y_i = \Lambda(t_i)$. But the condition $x_i = x_{i-1}$ is equivalent to $y_i = y_{i-1}$. Performing the transformation of variables we obtain (36).

The proof of the above proposition leads to a very simple extension of the algorithm of the previous section.

Theorem 4. Steps 1–6 of the algorithm of Section 1.2 are valid for time-varying $\lambda(t)$ if t is replaced by $\Lambda(t)$ (and t_j with $\Lambda(t_j)$).

Proof. This theorem follows immediately by applying the original proof of Steps 1–6 to the time-varying case and using the transformation of variables $y_i = \Lambda(x_i)$ as was done in the above proposition.

Remark. Note that the piecewise linearity property with respect to t of $E[Q(t) \mid O'(\vec{t}, n)]$ in (13) found for the case of constant λ is destroyed by the transformation of variables, i.e.,

$$E[Q(t) \mid O'(\vec{t}, n)] = (1 - \theta)E[N(t_{j-1}) \mid O'(\vec{t}, n)] + \theta E[N(t_j) \mid O'(\vec{t}, n)]$$

where

$$\theta = \frac{\Lambda(t) - \Lambda(t_{j-1})}{\Lambda(t_j) - \Lambda(t_{j-1})}.$$

Note that while the performance measures in the homogeneous Poisson case of the previous section do not depend on the knowledge of λ , the results in the time-varying case require knowledge of $\Lambda(t)$.

3. STATIONARY RENEWAL ARRIVAL PROCESS

Our methods in the previous sections critically depend on the Poisson assumption. In this section, this assumption is relaxed and we only assume that the arrival process is a renewal time-homogeneous process, where the probability density function between two successive arrivals is a known function $f(x)$

with cdf $F(x) = \int_{y=0}^x f(y) dy$. The key step is to generalize (2).

Theorem 5

$$\Pr\{X_2 \leq t_1, \dots, X_n \leq t_{n-1} \mid N(t_n) = n\} = (F^{(n-1)}(t_n) - F^{(n)}(t_n)) \int_{\lambda_2=0}^{t_1} \dots \int_{\lambda_n=\lambda_{n-1}}^{t_{n-1}} f(x_2)f(x_3 - x_2) \dots f(x_n - x_{n-1})(1 - F(t_n - x_n)) dx_2 \dots dx_n. \quad (38)$$

Proof. The underlying conditional density function can be expressed in terms of the ratio

$$h(X_2 = x_2, \dots, X_n = x_n \mid N(t_n) = n) = \frac{g\{X_2 = x_2, \dots, X_n = x_n, N(t_n) = n\}}{\Pr\{N(t_n) = n\}}.$$

But $g\{X_2 = x_2, \dots, X_n = x_n, N(t_n) = n\}$ is the joint density function corresponding to the first interarrival time being x_2 , the second interarrival time being $x_3 - x_2, \dots$, the $n - 1$ th interarrival being $x_n - x_{n-1}$, and having no arrival occurring in an interval of duration $t_n - x_n$. Hence,

$$g\{X_2 = x_2, \dots, X_n = x_n, N(t_n) = n\} = f(x_2)f(x_3 - x_2) \dots f(x_n - x_{n-1})(1 - F(t_n - x_n)).$$

Also,

$$\Pr\{N(t) = n\} = \Pr\{N(t) \geq n\} - \Pr\{N(t) \geq n + 1\} = \Pr\{X_n \leq t\} - \Pr\{X_{n+1} \leq t\} = F^{(n-1)}(t) - F^{(n)}(t),$$

where $F^{(n)}(t)$ is the cdf of the n th convolution of the renewal process and therefore the cdf of X_{n+1} . Then

$$h(X_2 = x_2, \dots, X_n = x_n \mid N(t_n) = n) = \frac{f(x_2)f(x_3 - x_2) \dots f(x_n - x_{n-1})(1 - F(t_n - x_n))}{F^{(n-1)}(t_n) - F^{(n)}(t_n)} \quad (39)$$

so (38) follows.

Equation 38 is the basis of the algorithm for estimating $Q(t)$ and $W(t)$. For example, using (38), one can evaluate

$$\Pr\{N(t_j) \geq k \mid O(\vec{t}), N(t_n) = n\} = (\Pr\{X_2 \leq t_1, \dots, X_{j+1} \leq t_j, \dots, X_k \leq t_j, X_{k+1} \leq t_k, \dots, X_n \leq t_{n-1} \mid N(t_n) = n\}) / (\Pr\{O(\vec{t}) \mid N(t_n) = n\}). \quad (40)$$

From (40), one can find

$$\begin{aligned} E[Q(t_j) | O(\vec{t}), N(t_n) = n] &= E[N(t_j) | O(\vec{t}), N(t_n) = n] - j \\ &= \sum_{k=1}^n \Pr\{N(t_j) \geq k | O(\vec{t}), N(t_n) = n\} - j \\ &= \sum_{k=j+2}^n \Pr\{N(t_j) \geq k | O(\vec{t}), N(t_n) = n\} + 1 \end{aligned}$$

because $\Pr\{N(t_j) \geq k | O(\vec{t}), N(t_n) = n\} = 1$ for $k = 1, 2, \dots, j + 1$.

For the Erlang- k distribution, for example, $f(x) = \lambda(\lambda x)^{k-1} e^{-\lambda x} / (k - 1)!$ and $F(x) = 1 - e^{-\lambda x} \sum_{j=0}^{k-1} (\lambda x)^j / j!$.

In this case, from (38), for all t_1, t_2, \dots, t_n ,

$$\begin{aligned} \Pr\{O(\vec{t}) | N(t_n) = n\} &= \Pr\{X_2 \leq t_1, \dots, X_n \leq t_{n-1} | N(t_n) = n\} \\ &= A_k(\lambda, t_n) \int_{\lambda_2=0}^{t_1} \dots \int_{\lambda_n=\lambda_{n-1}}^{t_{n-1}} x_2^{k-1} (x_3 - x_2)^{k-1} \\ &\quad \dots (x_n - x_{n-1})^{k-1} \sum_{j=0}^{k-1} \frac{(\lambda(t_n - x_n))^j}{j!} dx_2 \dots dx_n \quad (41) \end{aligned}$$

where

$$A_k(\lambda, t_n) = \frac{(\lambda^k / (k - 1)!)^n e^{-\lambda t_n}}{F^{(n-1)}(t_n) - F^{(n)}(t_n)}$$

Therefore, the evaluation of (40) can ignore the contribution of $A_k(\lambda, t_n)$ because it will cancel in the numerator and denominator. Note that for $k = 1$ we get the results of Section 1.

4. REAL-TIME ESTIMATES

In some applications, it is desirable not to wait until the end of a busy period to estimate the queue length. For example, if there is a possibility of real-time control of the service time, knowledge that the queue length was excessively large, but that it is currently zero, is not of value. Instead a current estimate is needed. The analysis of this problem will be derived for the case of a time-varying Poisson process. However, for the case of a constant arrival rate the method can easily be specialized.

Suppose that immediately after the n th departure (time t_n^+) we observe exactly n departures at times t_i , $i = 1, 2, \dots, n$ during a busy period, that the busy period did not end at time t_n (as inferred from the commencement of a new service initiation at time t_n). We are interested in having an estimate for the state

of the system at time $t \geq t_n$ without any observation of $D(t)$ between time t_n and time t . Also note that knowledge that the busy period did not end at time t_n is equivalent to $X_{n+1} \leq t_n$.

We first present the $O(n)$ on-line algorithm for estimating $N(t)$ and $Q(t)$ given these observations.

4.1. An Exact $O(n)$ On-Line Algorithm

Step 0. (Initialization). Let $g_0 = 1$; $t_0 = 0$.

Step 1. (On-line recursive update).

$$g_n = \sum_{i=1}^n (-1)^{n-i} \frac{\Lambda^{n-i+1}(t_i)}{(n-i+1)!} g_{i-1}, \quad (42)$$

$$R_n = R_{n-1} + T_{n-1} - [\Lambda(t_n) + 1]e^{-\Lambda(t_n)}g_{n-1}, \quad (43)$$

$$T_n = T_{n-1} - e^{-\Lambda(t_n)}g_{n-1}. \quad (44)$$

Step 2. (Real-time estimation)

$$E[Q(t_n) | O(\vec{t}), X_{n+1} \leq t_n] = 1 + \frac{\Lambda(t_n) - R_n}{T_n} \quad (45)$$

and for $t \geq t_n$,

$$E[N(t) | O(\vec{t}), X_{n+1} \leq t_n] = 1 + n + \frac{\Lambda(t) - R_n}{T_n}. \quad (46)$$

At each step we need to keep track of the vector g_i , $i = 1, \dots, n$, which requires $O(n)$ time. Using the same methodology we can recursively estimate the variance and higher moments of the queue length as well.

4.2. Proof of Correctness

In this subsection, we prove that indeed the algorithm correctly computes the distributions of the queue length and waiting time.

Theorem 6. For $t > t_n$, $E[Q(t_n) | O(\vec{t}), X_{n+1} \leq t_n]$ and $E[N(t) | O(\vec{t}), X_{n+1} \leq t_n]$ are given by (45) and (46), respectively, where g_n, R_n, T_n are computed recursively by (42), (43), and (44), respectively.

Proof. For $j \geq n + 1$, using (35), we obtain that

$$\begin{aligned} \Pr\{O(\vec{t}), X_{n+1} \leq t_n | N(t) = j\} &= \frac{(j-1)!}{\Lambda(t)^{j-1}} \int_{\lambda_2=0}^{t_1} \dots \int_{\lambda_n=\lambda_{n-1}}^{t_{n-1}} \int_{\lambda_{n+1}=x_n}^{t_n} \int_{\lambda_{n+2}=x_{n+1}}^t \\ &\quad \dots \int_{\lambda_j=\lambda_{j-1}}^t \prod_{i=2}^j \lambda(x_i) dx_i \\ &= \frac{(j-1)!}{\Lambda(t)^{j-1}} \int_{\lambda_2=0}^{t_1} \dots \int_{\lambda_n=\lambda_{n-1}}^{t_{n-1}} \int_{\lambda_{n+1}=x_n}^{t_n} \\ &\quad \cdot \frac{[\Lambda(t) - \Lambda(x_{n+1})]^{j-n-1}}{(j-n-1)!} \prod_{i=2}^{n+1} \lambda(x_i) dx_i \quad (47) \end{aligned}$$

where we used Lemma 1 and the transformation of variables $y_i = \Lambda(x_i)$, $i = n + 2, \dots, j$. Also,

$$\Pr\{N(t) = j\} = e^{-\lambda(t)} \frac{\lambda(t)^{j-1}}{(j-1)!}. \tag{48}$$

Now, for $t_n < t$, let

$$\begin{aligned} T_n &= \Pr\{O(\vec{t}), X_{n+1} \leq t_n\} \\ &= \sum_{j=n+1}^{\infty} \Pr\{O(\vec{t}), X_{n+1} \leq t_n | N(t) = j\} \Pr\{N(t) = j\} \\ &= \sum_{j=n+1}^{\infty} e^{-\lambda(t)} \int_{x_2=0}^{t_1} \dots \int_{x_n=x_{n-1}}^{t_{n-1}} \int_{x_{n+1}=x_n}^{t_n} \\ &\quad \cdot \frac{[\lambda(t) - \Lambda(x_{n+1})]^{j-n-1}}{(j-n-1)!} \prod_{i=2}^{n+1} \lambda(x_i) dx_i, \end{aligned} \tag{49}$$

from (47) and (48), which leads to

$$\begin{aligned} T_n &= \int_{x_2=0}^{t_1} \dots \int_{x_n=x_{n-1}}^{t_{n-1}} \int_{x_{n+1}=x_n}^{t_n} \\ &\quad \cdot e^{-\Lambda(x_{n+1})} \prod_{i=2}^{n+1} \lambda(x_i) dx_i. \end{aligned} \tag{50}$$

But, for $t > t_n$,

$$\begin{aligned} E[N(t) | O(\vec{t}), X_{n+1} \leq t_n] &= \sum_{j=n+1}^{\infty} j \Pr\{N(t) = j | O(\vec{t}), X_{n+1} \leq t_n\} \\ &= \sum_{j=n+1}^{\infty} (n+1 + (j-n-1)) \\ &\quad \cdot \frac{\Pr\{O(\vec{t}), X_{n+1} \leq t_n | N(t) = j\} \Pr\{N(t) = j\}}{\Pr\{O(\vec{t}), X_{n+1} \leq t_n\}}. \end{aligned}$$

From (47), (48) and (50) we find that this is equal to

$$\begin{aligned} &\frac{1}{T_n} \sum_{j=n+1}^{\infty} (n+1 + (j-n-1)) e^{-\lambda(t)} \\ &\int_{x_2=0}^{t_1} \dots \int_{x_n=x_{n-1}}^{t_{n-1}} \int_{x_{n+1}=x_n}^{t_n} \\ &\quad \frac{[\lambda(t) - \Lambda(x_{n+1})]^{j-n-1}}{(j-n-1)!} \prod_{i=2}^{n+1} \lambda(x_i) dx_i \\ &= \frac{1}{T_n} [(n+1)T_n + \int_{x_2=0}^{t_1} \dots \int_{x_n=x_{n-1}}^{t_{n-1}} \int_{x_{n+1}=x_n}^{t_n} \\ &\quad [\lambda(t) - \Lambda(x_{n+1})] e^{-\Lambda(x_{n+1})} \prod_{i=2}^{n+1} \lambda(x_i) dx_i]. \end{aligned}$$

Therefore, (46) follows where

$$\begin{aligned} R_n &= \int_{x_2=0}^{t_1} \dots \int_{x_n=x_{n-1}}^{t_{n-1}} \int_{x_{n+1}=x_n}^{t_n} \\ &\quad \cdot \Lambda(x_{n+1}) e^{-\Lambda(x_{n+1})} \prod_{i=2}^{n+1} \lambda(x_i) dx_i. \end{aligned}$$

After computing the innermost integral using integration by parts,

$$\begin{aligned} R_n &= \int_{x_2=0}^{t_1} \dots \int_{x_n=x_{n-1}}^{t_{n-1}} (-[\Lambda(t_n) + 1] e^{-\Lambda(t_n)} \\ &\quad + [\Lambda(x_n) + 1] e^{-\Lambda(x_n)}) \prod_{i=2}^n \lambda(x_i) dx_i, \end{aligned}$$

and thus (43) follows where

$$g_n = \int_{x_2=0}^{t_1} \dots \int_{x_{n+1}=x_n}^{t_n} \prod_{i=2}^{n+1} \lambda(x_i) dx_i. \tag{51}$$

Computing the innermost integral of (50), we get (44).

From (5) and (51) and the transformation of variables, $y_i = \Lambda(x_i)$ described in Section 2, one finds that the definition of g_n would equal $h_{n+1, n+1}$ if t_i is replaced by $\Lambda(t_i)$. Therefore, from (8), g_n satisfies (42).

Finally, $D(t_n) = n$ and $Q(t) = N(t) - D(t)$. Hence, (45) follows from (46).

Remark. To find $E[Q(t) | O(\vec{t}), X_{n+1} < t_n]$ for $t \geq t_n$ requires knowledge of the departure process, i.e., knowledge of the service distribution and the number of servers present.

5. COMPUTATIONAL RESULTS

We implemented the algorithm of Section 1.2 for the case of stationary Poisson arrivals on a SUN 3 workstation. Using a straightforward implementation, the algorithm uses n^2 memory (the matrices $f_{j,k}$ and $h_{j,k}$ are half full). For most practical applications, the number of departures in a busy period is less than 100. We were able to calculate several performance characteristics with up to $n = 99$ number of departures in a busy period reliably in less than 40 seconds. In Figure 1, we report the expected queue length of a busy period of 99 customers just after each departure epoch given that the departure epochs are regularly spaced within the busy period, i.e., $t_i = i/n$. In Figure 2, we compute the estimate of the average queue length during the entire busy period parameterized by n , the total number of arrivals during the busy period, based on the assumption that the departures

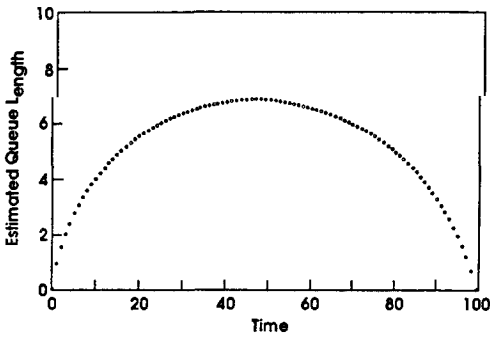


Figure 1. Estimated queue length versus time when $n = 99, t_i = i/n$.

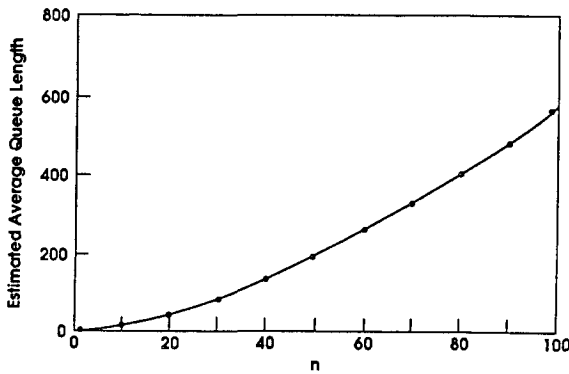


Figure 2. Estimated average queue length versus n when $t_i = i/n, n = 1, \dots, 99$.

are regularly spaced, i.e., given n we compute

$$\int_{t=0}^{t_n} E[Q(t) | O'(\vec{t}, n)] dt.$$

Not surprisingly, this is a monotonically increasing function of n .

For the case of $n = 5$, in Figure 3 we compute the estimate of the queue length just after departure epochs, $E[Q(t, +) | O'(\vec{t}, n)]$, based on the assumption that the departures are at $t_i = i/5, \lambda = 10$, and the interarrival time is either exponentially distributed or Erlang-2 distributed. The Erlang-2 distribution curve was based on the analysis in Section 3 using *Mathematica*. For graphical simplicity these points are connected. Of course, the queue length decreases by one at each departure epoch so that the continuous curves in Figure 3 should not be confused with the discontinuous function $E[Q(t) | O'(\vec{t}, n)]$. For the case of the exponential interarrival time, $E[Q(t) | O'(\vec{t}, n)]$ was shown to be the piecewise linear (cf. (13)) so $E[Q(t) | O'(\vec{t}, n)]$ could be reconstructed from

Figure 3. For the Erlang-2 interarrival time case no piecewise linearity has been established so a similar reconstruction $E[Q(t) | O'(\vec{t}, n)]$ from Figure 3 could only be viewed as an approximation. As expected, the larger coefficient of variation of the exponential distribution causes a larger queue length. Note also that for the exponential case the curve is convex as anticipated in Larson (1990). On the other hand, the Erlang-2 curve is clearly not convex.

If all the parameters of Figure 3 remain unchanged except that λ is increased to 100, then the exponential

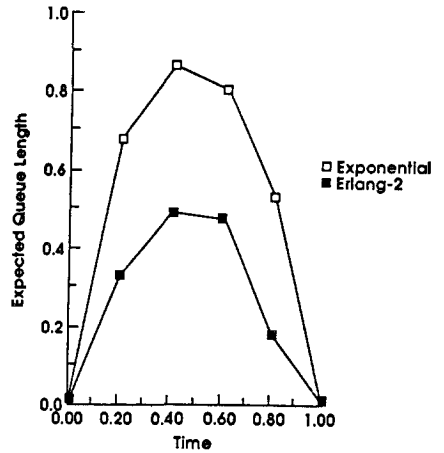


Figure 3. Expected queue length versus time for the interarrival distribution: exponential or Erlang-2, $n = 5, t_i = i/5, \lambda = 10$.

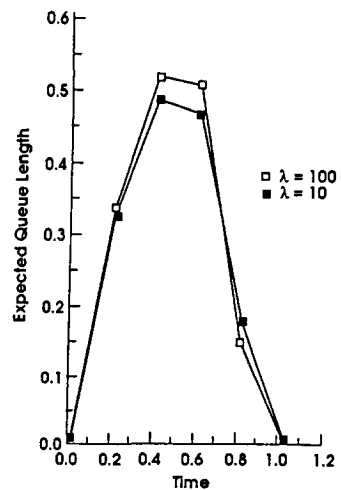


Figure 4. Expected queue length versus time for the interarrival distribution: Erlang-2, $n = 5, t_i = i/5, \lambda = 10$ or 100.

case curve would not change. This is because the algorithm in Section 1 is independent of λ . Figure 4 compares $\lambda = 10$ with $\lambda = 100$ for the case of Erlang-2 interarrival times and again plots a connection of the points $E[Q(t, +) | O'(t, n)]$. It is interesting to note that the two curves are quite close.

ACKNOWLEDGMENT

The research of the first author was partially supported by the International Financial Services Center and the Leaders of Manufacturing Program at MIT. The research of the second author was partially supported by NSF grant ECES-88-15449 and U. S. Army con-

tract DAAL-03-83-K-0171 as well as the Laboratory for Information and Decision Sciences (LIDS) at MIT.

REFERENCES

- GARLICK, R. 1990. Estimating Disperse Network Queues: The Queue Inference Engine. *Computer Comm. Rev.* **20**, 111-118.
- LARSON, R. 1990. The Queue Inference Engine: Deducing Queue Statistics From Transactional Data. *Mgmt. Sci.* **36**, 586-601.
- LARSON, R. 1989. The Queue Inference Engine (QIE). CORS/TIMS/ORSA Joint National Meeting, Vancouver, Canada.
- ROSS, S. 1983. *Stochastic Processes*. John Wiley, New York.