AT&T Bell Laboratories Technical Journal
Vol. 63, No. 7, September 1984
Printed in U.S.A.

Methods in Traffic Calculations

By D. L. JAGERMAN*

(Manuscript received February 24, 1984)

This paper presents practical formulae and computer programs for the following traffic-related problems: determination of offered load; determination of number of trunks; equivalent random method and Hayward's method; the three-moment match (construction of interrupted Poisson stream); day-to-day load variation, including both time and call congestions; and the analysis of a multiserver queue in a traffic environment.

I. INTRODUCTION

This paper presents methods of traffic calculations that have been worked out and modified by the author for the typical range of problems encountered in traffic applications. Their efficacy depends, in large part, on improved computations of the Erlang loss function and its derivatives. Practical computer programs are given in the Appendix for immediate application. These programs were originally written by the author in TI EXTENDED BASIC but were transcribed to Fortran by Brian Farrell. All of the programs except for the GT/M/S queue were also written in POCKET BASIC for execution on the TRS-80† PC1. The programs are simple and are executed rapidly.

This paper does not fully develop the theories of all of the formulae used. Instead, it gives enough description and explanation to make the development of the computational formulae understandable. Numerical examples show the operation of the methods and programs. These results are intended to serve only as checks on the computer programming; for theoretical accuracy, one should consult the references.

* AT&T Bell Laboratories.

† Trademark of Tandy Corporation.

Traffic theory traditionally was concerned only with blocking phenomena. However, modern systems are often analyzed with respect to delays, which arise because of the complex implementation of the communication system and the increasing range of services being provided. For example, delays occurring in systems that yield a response to a query (reservation systems, credit card systems, etc.) are important enough to require careful analysis. In this paper a multiserver delay queue GT/M/S, whose arrival stream is typical of those occurring in traffic theory, is analyzed using the tools of that theory, which are developed here. The waiting time distribution and mean waiting time may be conveniently calculated using the program for the GT/M/S queue given in the Appendix.

II. INTERPOLATION

The Erlang loss function arises in the study of the M/M/n/n queueing problem, that is, a Poisson stream of calls offering a erlangs to a fully available trunk group of n trunks; the Erlang loss function expresses the blocking probability, that is, the probability that an arriving call is rejected because no trunk is available. It usually is stated in the form[1]

$$B(n, a) = \frac{a^n}{n!} \Big/ \sum_{j=0}^{n} \frac{a^j}{j!} \tag{1}$$

or, equivalently,

$$B(n, a)^{-1} = \sum_{j=0}^{n} \frac{n!}{j!} a^{-(n-j)}. \tag{2}$$

In terms of the descending factorial, $n^{(j)}$, defined by

$$n^{(0)} = 1, \qquad n^{(j)} = n(n-1) \cdots (n-j+1)(j \geq 1), \tag{3}$$

one may also write

$$B(n, a)^{-1} = \sum_{j=0}^{n} n^{(j)} a^{-j}. \tag{4}$$

The integral representation of Fortet[2] may be obtained from (4) as follows. From the Eulerian integral

$$a^{-j} = a \int_0^\infty e^{-ay} \frac{y^j}{j!} dy, \tag{5}$$

one has

$$B(n, a)^{-1} = a \int_0^\infty e^{-ay} \sum_{j=0}^{n} \frac{n^{(j)}}{j!} y^j dy \tag{6}$$

and, hence

$$B(n, a)^{-1} = a \int_0^\infty e^{-ay}(1 + y)^n dy. \qquad (7)$$

Since the above integral has meaning for nonintegral n, the general definition of $B(x, a)$, in which $x > 0$ is unrestricted and $a > 0$ is (see Ref. 3)

$$B(x, a)^{-1} = a \int_0^\infty e^{-ay}(1 + y)^x dy. \qquad (8)$$

This relates $B(x, a)$ to the incomplete gamma function,[4] namely,

$$B(x, a)^{-1} = a^{-x}e^a \int_a^\infty e^{-y}y^x dy, \qquad (9)$$

$$= a^{-x}e^a\Gamma(x + 1, a). \qquad (10)$$

The function $B(x, a)^{-1}$ satisfies an important linear difference equation that will now be obtained. Integration by parts applied to (8) yields

$$B(x, a)^{-1} = 1 + x \int_0^\infty e^{-ay}(1 + y)^{x-1} dy. \qquad (11)$$

Hence,

$$B(x, a)^{-1} = \frac{x}{a} B(x - 1, a)^{-1} + 1. \qquad (12)$$

This is an excellent recursion for the successive computation of $B(x, a)^{-1}$. For integral values of x, the initial value $B(0, a) = 1$ is convenient.

For use in the equivalent random method[5] and Hayward's approximation,[6] to be discussed later, it is important to have an easily calculable approximation for $B(x, a)$ when x is nonintegral. For this purpose Newton's interpolation formula will be used.[7] Define the forward difference operator, Δ, by

$$\Delta f(x) = f(x + 1) - f(x). \qquad (13)$$

Then, the powers $\Delta^2, \Delta^3, \cdots$, are defined by successive application of Δ, and thus

$$\Delta^2 f(x) = f(x + 2) - 2f(x + 1) + f(x). \qquad (14)$$

Newton's interpolation formula is

$$f(x + h) = \sum_{j=0}^{\infty} \binom{h}{j} \Delta^j f(x). \qquad (15)$$

Let

$$f(x) = \ln B(x, a), \tag{16}$$

$$n = [x], \qquad h = x - n, \tag{17}$$

in which the brackets designate integral part. Newton's formula is now applied up to the second difference to obtain

$$\ln B(x, a) \simeq \ln B(n, a) + h\Delta \ln B(n, a) + \frac{1}{2} h(h - 1)\Delta^2 \ln B(n, a). \tag{18}$$

For convenience in writing, the following abbreviations are used

$$B = B(n, a), \qquad B_1 = B(n + 1, a), \qquad B_2 = B(n + 2, a). \tag{19}$$

One now obtains, from (18),

$$B(x, a) \simeq B^{1-h} B_1^h \left[\frac{B_1^2}{BB_2}\right]^{\frac{1}{2}h(1-h)}. \tag{20}$$

The accuracy of (20) improves with increasing x. The worst error occurs at $h = 0.5$. Some comparisons are given in Table I.

III. DERIVATIVES

For economic considerations and for iteration formulae for the solution of equations involving $B(x, a)$, as exemplified later in this paper, the derivatives $\partial B(x, a)/\partial a = B_a$ and $\partial B(x, a)/\partial x = B_x$ are needed.[8] The symbol $B = B(x, a)$ will be used.

From (8) differentiation with respect to a yields

$$-B^{-2}B_a = \int_0^\infty e^{-ay}(1 + y)^x dy - a \int_0^\infty e^{-ay} y(1 + y)^x dy, \tag{21}$$

$$= \int_0^\infty e^{-ay}(1 + y)^x dy - a \int_0^\infty e^{-ay}(1 + y)^{x+1} dy + a \int_0^\infty e^{-ay}(1 + y)^x dy, \tag{22}$$

Table I—Comparison of interpolation with exact values

x	a	$B=$	$B\simeq$
1.5	0.1	0.02155	0.02132
5.5	2	0.02146	0.02145
10.5	8	0.10011	0.10013
100.5	90	0.02517	0.02517

$$= \left(1 + \frac{1}{a}\right) B^{-1} - B(x + 1, a)^{-1}, \tag{23}$$

$$B_a = \left(\frac{x + 1}{a} B^{-1} + 1\right) B^2 - \left(1 + \frac{1}{a}\right) B, \tag{24}$$

$$B_a = \left(\frac{x}{a} - 1 + B\right) B. \tag{25}$$

For B_x one has, from (8),

$$B_x = -B^2 a \int_0^\infty e^{-ay}(1 + y)^x \ln(1 + y) dy. \tag{26}$$

Unfortunately, there is no exact evaluation of (26) in convenient form. Useful, easily calculable approximations may, however, be obtained. First, a crude but useful approximation will be obtained. Let

$$f(y) = Bae^{-ay}(1 + y)^x. \tag{27}$$

Then, from (8),

$$f(y) \geqslant 0, \qquad \int_0^\infty f(y) dy = 1. \tag{28}$$

Also, one has

$$\int_0^\infty y f(y) dy = B(x + 1)^{-1} B - 1, \tag{29}$$

$$= \frac{x + 1}{a} + B - 1. \tag{30}$$

Jensen's inequality[9] for a random variable (ξ) and a function $g(x)$ convex on the range of (ξ) is

$$Eg(\xi) \geqslant g(E\xi). \tag{31}$$

Accordingly, let ξ have the density function $f(y)$. Then, (26) may be expressed as

$$B_x/B = -E \ln(1 + \xi). \tag{32}$$

Since $-\ln(1 + y)$ is convex on $y \geqslant 0$, using Jensen's inequality gives

$$B_x/B \geqslant -\ln\left(\frac{x + 1}{a} + B\right), \tag{33}$$

in which $E\xi$ was obtained from (30). It will be convenient to set

$$\alpha = \frac{x + 1}{a} + B \tag{34}$$

so that one has

$$B_x/B \geqslant -\ln \alpha. \tag{35}$$

To obtain a better approximation to B_x than the lower bound of (33), the difference equation (12) will be used. Again, let

$$f(x) = \ln B(x, a).$$

Then, the difference equation becomes

$$f(x + 1) - f(x) = -\ln \alpha. \tag{36}$$

The Taylor expansion for $f(x + 1) - f(x)$ gives

$$f'(x) \cong -\ln \alpha - \frac{1}{2} f''(x). \tag{37}$$

From $f'(x) \cong -\ln \alpha$, one has $f'' \cong -\frac{\alpha'}{\alpha}$; hence,

$$f'(x) \cong -\ln \alpha + \frac{\alpha'}{2\alpha}. \tag{38}$$

One has

$$f'(x) = \frac{B_x}{B}, \tag{39}$$

$$\alpha' = \frac{1}{a} + B_x. \tag{40}$$

Substituting these values of f', α' into (38) yields the following approximation for B_x:

$$B_x/B \cong -\frac{\ln \alpha - 1/(2a\alpha)}{1 - B/(2\alpha)}. \tag{41}$$

With (35) designated as bound and (41) designated as approximation, Table II presents comparisons with exact values taken from the table of Akimaru and Nishimura. Throughout the table $B = 0.01$.

IV. DETERMINATION OF OFFERED LOAD

In the equation

$$B(x, a) = P, \tag{42}$$

Table II—Comparisons of derivative values

x	a	$-B_x/B$	Bound	Approximation
5	1.3608	1.4025	1.4860	1.4044
10	4.4612	0.8626	0.9065	0.8630
20	12.0306	0.5406	0.5628	0.5406
50	37.9014	0.2956	0.3042	0.2956

in which x and P are given, a is to be determined. Newton's method of iteration is well suited to the problem. Let a_0 be an initial approximation, and let a_1 be the refined result. Then,

$$a_1 = a_0 - \frac{B_0 - P}{(B_a)_0}, \tag{43}$$

in which the subscript 0 indicates evaluation at $a = a_0$. Using (25) one has

$$a_1 = a_0 - \frac{B_0 - P}{\left(\dfrac{x}{a_0} - 1 + B_0\right) B_0}. \tag{44}$$

The problem of obtaining a good starting point remains. For this purpose an inequality for B is obtained.

From

$$(1 + y)^x \leqslant e^{xy} \tag{45}$$

and (8) one gets

$$B(x, a)^{-1} \leqslant \frac{a}{a - x}, \qquad x < a. \tag{46}$$

Using (46) in (12) now yields

$$B(x, a) \geqslant 1 - \frac{x}{a + 1}, \tag{47}$$

with no restriction on x. Thus, setting $B = P$ in (47), the initial value can be

$$a_0 = \frac{x}{1 - P} - 1. \tag{48}$$

As an example of the convergence rate of (44) starting with (48), consider $x = 20$, $P = 0.01$. The following values were obtained using the program given in the Appendix.

$$a_0 = 19.202, \qquad a_1 = 14.057, \qquad a_2 = 12.568,$$

$$a_3 = 12.088, \qquad a_4 = 12.031, \qquad a_5 = 12.031.$$

Another important case occurs when the carried load, L, is specified and the offered load is required; the relevant equation is

$$L = a(1 - B(x, a)). \tag{49}$$

Newton's formula in the form

$$a_1 = a_0 - \frac{a_0(1 - B_0) - L}{(L_a)_0} \tag{50}$$

is used. Since

$$L_a = 1 - B - aB_a, \tag{51}$$

one has, using (25) and (5),

$$a_1 = a_0 - \frac{a_0(1 - B_0) - L}{1 - B_0 - (x - a_0 + a_0B_0)B_0}. \tag{52}$$

The inequality

$$a < L\left(1 + \frac{L}{x(x - L)}\right) \tag{53}$$

gives a convenient starting value for (52); thus,

$$a_0 = L\left(1 + \frac{L}{x(x - L)}\right). \tag{54}$$

The convergence rate is the same as in (44).

V. DETERMINATION OF NUMBER OF TRUNKS

The equation

$$B(x, a) = P \tag{55}$$

will now be solved for x given a and P. The Newton iteration formula now reads

$$x_1 = x_0 - \frac{B_0 - P}{(B_x)_0}. \tag{56}$$

The bound for B_x, namely (33), will be used in (56); thus

$$x_1 = x_0 + \frac{B_0 - P}{B_0 \ln \alpha_0}. \tag{57}$$

The starting value for x is again obtained from (47); it is

$$x_0 = (1 - P)(1 + a). \tag{58}$$

One must use (20) to evaluate B, since x need not be an integer. Using the approximate value (33) for B_x does not impair the accuracy of the result. It merely slows down the convergence rate over what Newton's method would provide with the exact derivative.

Using the program of the appendix and $a = 12.031$, $P = 0.01$, the following values are obtained:

$x_0 = 12.901$, $x_1 = 16.319$, $x_2 = 18.352$, $x_3 = 19.505$,

$x_4 = 19.932$, $x_5 = 19.997$, $x_6 = 20.000$.

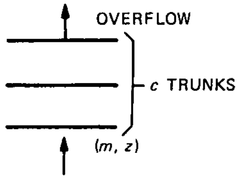

Fig. 1—Common trunk group.

VI. EQUIVALENT RANDOM METHOD AND HAYWARD'S METHOD

Figure 1 schematizes the problem of ascertaining blocking on a common trunk group. The common trunk group is offered a composite stream, which is the superposition of overflow streams.

The input stream is characterized by the offered load m and peakedness z. The equivalent random method considers the stream (m, z) to be the overflow stream of a fictitious trunk group of x trunks and Poisson-offered load a. This is shown in Fig. 2.

If the stream (m, z) is not in fact the overflow of a single trunk group, then x need not be an integer. The Kosten formulae for overflow and peakedness are

$$m = aB(x, a), \tag{59}$$

$$z = 1 - m + \frac{a}{x + m + 1 - a}. \tag{60}$$

These can be arranged in the following form:

$$m = aB\left[a\frac{m + z}{m + z - 1} - m - 1, a\right], \tag{61}$$

$$x = a\frac{m + z}{m + z - 1} - m - 1. \tag{62}$$

The problem is to determine the equivalent random parameters x and

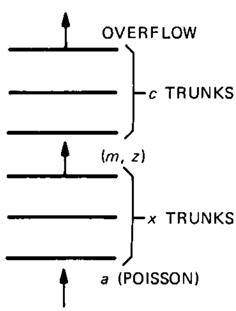

Fig. 2—Common and fictitious trunk groups.

a; that is, to solve (61) for a and then obtain x from (62). When x and a are known then, of course, the required blocking probability, P, is given by

$$P = \frac{B(x + c, a)}{B(x, a)}. \tag{63}$$

Newton's method will be used to solve for a. Thus,

$$a_1 = a_0 - \frac{aB_0 - m}{\frac{d}{da}(aB)_0}, \tag{64}$$

in which $B = B\left(a \frac{m + z}{m + z - 1} - m - 1, a\right)$.

One has

$$\frac{d}{da}(aB) = B + (x + m + 1)B_x + aB_a, \tag{65}$$

and using (25) and (33),

$$\frac{d}{da}(aB) \cong [x + m + 1 - a - (x + m + 1)\ln \alpha]B. \tag{66}$$

Newton's eq. (64) now becomes

$$a_1 = a_0 - \frac{a_0B_0 - m}{[x_0 + m + a - a_0 - (x_0 + m + 1)\ln \alpha_0]B_0}. \tag{67}$$

The formula of Rapp,[10] namely,

$$a_0 = mz + 3z(z - 1), \tag{68}$$

will start the iteration.

The following test case was used on the program in the Appendix. For $x = 10$, $a = 8$ one finds, from (59) and (60), that $m = 0.97329$, $z = 2.04016$. The results of the run are

$$x_0 = 10.527, \quad x_1 = 10.181, \quad x_2 = 10.067,$$
$$a_0 = 8.352, \quad a_1 = 8.121, \quad a_2 = 8.045,$$
$$x_3 = 10.025, \quad x_4 = 10.010, \quad x_5 = 10.004,$$
$$a_3 = 8.017, \quad a_4 = 8.007, \quad a_5 = 8.003.$$

The equivalent random method is usually used only when the service distribution on the common group is exponential, since the Kosten formula was derived for that distribution. However, in this regard, see Ref. 11 for a discussion of the constant service time case. To consider blocking on a common group with other service distributions, the approximation of W. S. Hayward is used. (See Ref. 6 for example.) In Fig. 1 the approximation is

$$P_h = B\left(\frac{c}{z}, \frac{m}{z}\right). \tag{69}$$

To use (69), the peakedness, z, must be referred to the service distribution that is considered. This will now be discussed.[12] For a service distribution, $F(x)$, with service rate μ, let

$$F_0(x) = F(x/\mu), \tag{70}$$

that is, the distribution scaled to unit rate, and further

$$F_0^c(x) = 1 - F(x),$$

$$F_0^{c,(2)}(y) = \int_0^\infty F_0^c(x)F_0^c(x+y)dx. \tag{71}$$

The notation $z(F_0^c;\, \mu)$ will be used to show the dependence of z on μ as a function and on F as a functional. Since z is usually known relative to some distribution, it would be useful to be able to transform z to other distributions. The Mellin transform will accomplish this.

For a function, $f(x)$, defined on $(0, \infty)$, the function $\bar{f}(s)$ defined by

$$\bar{f}(s) = \int_0^\infty x^{s-1}f(x)dx \tag{72}$$

is called the Mellin transform of $f(x)$. Let

$$f(\mu) = z(F_0^c;\, \mu) - 1, \tag{73}$$

$$g(\mu) = z(G_0^c;\, \mu) - 1. \tag{74}$$

Then, the required transformation formula for z is

$$\bar{g}(s) = \frac{\bar{G}_0^{c,(2)}(s)}{\bar{F}_0^{c,(2)}(s)}\bar{f}(s). \tag{75}$$

For a renewal stream with renewal density $m(\tau)$, the peakedness may be calculated directly by

$$z(F_0^c;\, \mu) = 1 - \frac{\lambda}{\mu} + 2\mu\int_0^\infty F^{c,(2)}(u)m(u)du. \tag{76}$$

As an example, consider the stream given by

$$m(\tau) = \lambda + Ae^{-\alpha\tau}. \tag{77}$$

By (76), z relative to exponential service is

$$z(e^{-x};\,\mu) = 1 + \frac{A}{\alpha + \mu}. \tag{78}$$

It is desired to transform this to the service distribution

$$G(x) = 1 - (x + 1)e^{-x}, \tag{79}$$

for which $\mu = 1/2$. One has

$$\bar{F}_0^{c,(2)}(s) = \frac{1}{2}\,\Gamma(s), \tag{80}$$

$$\bar{G}_0^{c,(2)}(s) = \frac{5}{8}\,2^{-s}\Gamma(s) + \frac{3}{4}\,2^{-s-1}\Gamma(s+1). \tag{81}$$

Hence,

$$\frac{\bar{G}_0^{c,(2)}(s)}{\bar{F}_0^{c,(2)}(s)} = \frac{5}{4}\,2^{-s} + \frac{3}{4}\,2^{-s}s. \tag{82}$$

Since

$$\bar{f}(s) = \frac{\pi}{\sin \pi s}\,A\alpha^{s-1}, \tag{83}$$

one now obtains from (75) and (82),

$$\bar{g}(s) = \frac{\pi}{\sin \pi s}\,A\left[\frac{5}{8}\left(\frac{\alpha}{2}\right)^{s-1} + \frac{3}{8}\left(\frac{\alpha}{2}\right)^{s-1} s\right]. \tag{84}$$

Hence,

$$z(G_0^c;\,\mu) = 1 + \frac{A}{8}\left[\frac{10}{\alpha + 2\mu} + \frac{12\mu}{(\alpha + 2\mu)^2}\right]. \tag{85}$$

Since $\mu = 1/2$, one finally has

$$z = 1 + \frac{A}{8}\left[\frac{10}{\alpha + 1} + \frac{6}{(\alpha + 1)^2}\right]. \tag{86}$$

Consider the following numerical examples. Let the common group have exponential service distribution, and let $c = 15$, $m = 10$, $z = 3$. Then, the equivalent random parameters are $x = 39.615$, $a = 46.721$. Thus, one has

$$P = \frac{B(54.615,\, 46.721)}{B(39.615,\, 46.721)} = 0.151, \tag{87}$$

$$P_h = B(5,\, 3.333) = 0.139. \tag{88}$$

Let the service rate for this example be $\mu = 1$, and let the arrival stream be that defined by (77) with $\lambda = 10$, $A = 4$, $\alpha = 1$. This is consistent with $m = 10$, $z = 3$. Now let the service distribution be that of (79). Then, by (86)

$$z = 4.25 \tag{89}$$

and

$$P_h = B\left(\frac{15}{4.25}, \frac{10}{4.25}\right) = 0.187. \tag{90}$$

VII. THE THREE-MOMENT MATCH

The interrupted Poisson process is on (flows) for an exponential period of time whose mean duration is γ^{-1}, and is off (stopped) for an exponential period whose mean duration is ω^{-1}. This may be thought of as a Poisson process of rate λ entering a switch that is alternately closed and opened. The output of the switch is the interrupted Poisson process with a rate λ'. This is shown in Fig. 3.

A stream that is the overflow of a single trunk group with Poisson offered load will be called on 0-stream. The interrupted Poisson process provides a useful approximation to an 0-stream.[13] The technique of approximation offers the 0-stream to an infinite server group and offers the interrupted Poisson stream to another infinite server group. In each group the distribution of the number of busy servers at any instant of time is obtained. The first three moments of these distributions are then equated. They give equations that define the interarrival time distribution of the interrupted Poisson stream. Since this stream is renewal, it is now completely defined.

Let the 0-stream be the overflow from a trunk group with x trunks and offered load a, and let

$$B = B(x, a), \qquad B_1 = B(x+1, a), \qquad B_2 = (x+2, a). \tag{91}$$

Then, the required equations for λ, λ', and the switch parameters, γ, ω, are

$$\delta_0 = B, \qquad \delta_1 = \frac{1}{a}\frac{\delta_0^{-1}}{B_1^{-1} - B^{-1}} \tag{92}$$

$$\delta_2 = \frac{2}{a^2}\frac{\delta_0^{-1}\delta_1^{-1}}{B_2^{-1} - 2B_1^{-1} + B^{-1}} \tag{93}$$

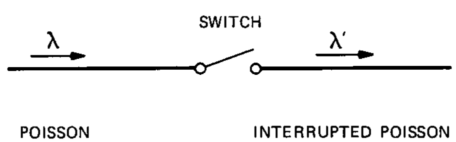

Fig. 3—Generation of interrupted Poisson process.

$$\lambda = a\,\frac{\delta_2(\delta_1 - \delta_0) - \delta_0(\delta_2 - \delta_1)}{2\delta_1 - \delta_0 - \delta_2}, \tag{94}$$

$$\omega = \frac{\delta_0}{\lambda}\,\frac{\lambda - a\delta_1}{\delta_1 - \delta_0} \tag{95}$$

$$\gamma = \frac{\omega}{a}\,\frac{\lambda - a\delta_0}{\delta_0}, \tag{96}$$

$$\lambda' = \lambda\,\frac{\omega}{\gamma + \omega}. \tag{97}$$

The Laplace transform, $\tilde{f}(s)$, of the interarrival time density function, $f(x)$, is

$$\tilde{f}(s) = \frac{\lambda s + \lambda\omega}{s^2 + (\lambda + \gamma + \omega)s + \lambda\omega}, \tag{98}$$

and the peakedness relative to exponential service is

$$z(\mu) = 1 + \frac{\lambda\gamma}{\gamma + \omega}\,\frac{1}{\gamma + \omega + \mu}. \tag{99}$$

The equations for the three-moment match were set up in Ref. 13, taking $\mu = 1$. Thus, (99) yields the exact peakedness of the 0-stream for $\mu = 1$. A program for the evaluation of λ, λ', γ, ω, is given in the Appendix.

The following numerical example was evaluated. For $x = 10$, $a = 8$, one finds $\lambda = 5.4405$, $\gamma = 2.7054$, $\omega = 0.5894$, $\lambda' = 0.9733$, and $z = 2.0402$.

The interrupted Poisson process is also used to construct a stream with given parameters (m, z) $(z > 1)$. The equivalent random parameters x, a are first found, then the above equations are used, even when x is not an integer, to obtain the parameters for the interrupted Poisson stream. As an example, consider the case used earlier for which $m = 10$, $z = 3$, with unit service rate. Using $x = 39.6148$, $a = 46.7214$, one finds $\lambda = 25.9889$, $\gamma = 4.3033$, $\omega = 2.6912$.

An interrupted Poisson stream may be used in simulation studies. It may also be used to obtain the blocking probability of Fig. 1, instead of the equivalent random method or Hayward's method. The blocking probability, P_{ip}, is given by[14]

$$P_{ip} = 1 + \sum_{j=1}^{c} C^{(j)}\lambda^{-j} \prod_{i=1}^{j} \frac{\gamma + \omega + i}{\omega + i}. \tag{100}$$

Since this process is constructed to be a good approximation to an 0-stream, it is to be expected that P_{ip} in (100) should be in close agreement with (63), as given by the equivalent random method. The following numerical examples of Table III show this.

The interrupted Poisson process is used when analyzing the delay queue considered in Section IX.

VIII. DAY-TO-DAY LOAD VARIATION

Observation of traffic on a consistent hour basis from day to day shows variation. To partially account for this variation,[5] one often assumes that the offered load, a, is random with a gamma density $f(x)$. Thus,

$$f(x) = e^{-x/\gamma} \frac{x^{\alpha-1}}{\Gamma(\gamma)\gamma^{\alpha}}, \tag{101}$$

$$\alpha\gamma = m, \qquad \alpha\gamma^2 = \sigma^2, \tag{102}$$

in which m, σ^2 are the mean and variance, respectively, of a. Any load-dependent statistic such as $B(x, a)$, $aB(x, a)$, z, etc. is replaced by its respective means. Let $g(a)$ be such a statistic; then it is necessary to evaluate $Eg(a)$. One can construct an approximation by using the Gauss-Laguerre quadrature theory.[15] Then the offered load can be considered to consist of two Poisson streams whose offered loads are a_1, a_2 and to occur with probabilities P_1, P_2, respectively. The quantities P_1, a_1, P_2, a_2 in terms of m and σ are

$$P_1 = \frac{1}{2} - \frac{1}{2} \frac{\sigma}{\sqrt{m^2 + \sigma^2}}, \tag{103}$$

$$a_1 = \frac{m^2 + \sigma^2 + \sigma\sqrt{m^2 + \sigma^2}}{m}, \tag{104}$$

$$P_2 = \frac{1}{2} + \frac{1}{2} \frac{\sigma}{\sqrt{m^2 + \sigma^2}}, \tag{105}$$

$$a_2 = \frac{m^2 + \sigma^2 - \sigma\sqrt{m^2 + \sigma^2}}{m}. \tag{106}$$

Thus, one has

$$Eg(a) \cong P_1 g(a_1) + P_2 g(a_2). \tag{107}$$

Table III—Comparison of interrupted Poisson method with equivalent random method

c	10	10	15
m	8	15	10
z	10	3	3
P_{ip}	0.5273	0.4910	0.1492
P	0.5278	0.4912	0.1507

For example, to compute day-to-day time congestion, one uses $EB(x, a)$.

Using the symbol $\bar{B}$ for this quantity, one has, from (107),

$$\bar{B} \cong P_1 B(x, a_1) + P_2 B(x, a_2). \tag{108}$$

Of possibly greater importance is the probability, P_B, that a call is blocked. To obtain this, let

$$0(x, a) = aB(x, a), \tag{109}$$

that is, $0(x, a)$ is the overflow traffic, and let $E0(x, a) = \bar{0}$, then

$$P_B = \frac{\bar{0}}{m}. \tag{110}$$

Thus, one has approximately

$$P_B \cong \frac{1}{m} \left[P_1 a_1 B(x, a_1) + P_2 a_2 B(x, a_2) \right]. \tag{111}$$

Wilkinson found empirically that one may often relate σ^2 to m through

$$\sigma^2 = 0.13m^{1.5}, \qquad 0.13m^{1.7}, \qquad 0.13m^{1.84} \tag{112}$$

for low, medium, and high variation, respectively.[5] Programs for evaluating $\bar{B}$ and P_B are given in the Appendix. A numerical example is $x = 20$, $m = 15$, and (112) for medium variation. ($\sigma^2 = 12.9807$.) One finds $P_B \cong 0.0795$ and $\bar{B} \cong 0.0635$.

The behavior of $\bar{B}$ is somewhat counter-intuitive, since, while $\bar{B}$ may at first increase with increasing σ, ultimately it decreases to zero.[16] The behavior of P_B is more satisfactory since it increases monotonically to one for $\sigma \rightarrow \infty$. One also has $P_B \geqslant \bar{B}$. An example for which $\bar{B} < B(x, m)$ is given by $B(3, 2) = 0.2105$, while $\bar{B} = 0.1988$ when $\sigma = 1$.

IX. GT/M/S QUEUE

Until now no delay queues were considered; however, it is becoming more and more important to estimate delays occurring in traffic systems. The approximations developed in this paper serve this purpose. In particular, the interrupted Poisson stream can be used as input to a GT/M/S queue for which an exact solution can be obtained. The peculiarity of the problem considered here is that the input stream is defined through specification of (m, z) rather than the usual specifications used in queueing theory (distribution of time between arrivals, etc.), hence the designation T for traffic. Considered from this point of view, a queue may be a queue in a traffic environment. Figure 4 shows this. For a discussion of this approach see Heffes.[17]

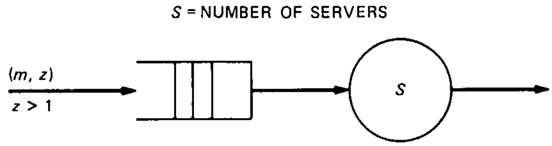

Fig. 4—Traffic-type queueing system.

The solution of the GT/M/S system with interrupted Poisson stream (obtained from three-moment match) as input may be expected to provide a close approximation when the input to the GT/M/S is a superposition of 0-streams. But even in more general situations the approximation is good.

The servers are assumed to have exponential service-time distribution with unit mean service rate, and to be identical and independent. The paradigm for the solution is as follows:

$$(m, z) \rightarrow (x, a) \rightarrow (\lambda, \gamma, \omega) \rightarrow w(t). \tag{113}$$

Thus, from the traffic, m, and peakedness, z, one obtains the equivalent random parameters x (number of trunks) and a (offered load), which give the interrupted Poisson parameters λ, γ, and ω. The interrupted Poisson stream is now used as input to a GI/M/S queue from which the exact waiting time distribution is computed. To accomplish the last step, the following formulae are used.[18] The function $\tilde{f}(s)$ is given in (98). Define r to be the root of

$$r = \tilde{f}[(1 - r)S], \tag{114}$$

where S is the number of servers, satisfying $0 < r < 1$, and $\delta = 1 - r$. Thus,

$$\delta = \frac{1}{2S^2}\left[\sqrt{\{(\lambda + \gamma + \omega)S - S^2\}^2 + 4S^2\{(\gamma + \omega)S - \lambda\omega\}} - \{(\lambda + \gamma + \omega)S - S^2\}\right]. \tag{115}$$

Define the renewal density, $\tilde{m}(s)$, by

$$\tilde{m}(s) = \frac{1}{1 - \tilde{f}(s)} - 1. \tag{116}$$

Then, the following quantities are calculated:

$$C_j = \prod_{i=1}^{j} \tilde{m}(i), \tag{117}$$

$$A^{-1} = \frac{1}{\delta} + \sum_{j=1}^{s} \frac{\binom{S}{j}}{C_j[1 - \tilde{f}(j)]} \frac{S[1 - \tilde{f}(j)] - j}{S\delta - j}, \tag{118}$$

$$P = \frac{A}{\delta}. \tag{119}$$

One now has

$$P[w > 0] = P, \tag{120}$$

$$P[w > t] = Pe^{-\delta St}, \tag{121}$$

$$EW = \frac{P}{\delta D}. \tag{122}$$

The above computation of delay is very sensitive to errors in the determination of $\tilde{f}(s)$, which, in turn, rests on an accurate evaluation of $B(x, a)$. Accordingly, another method of computing $B(x, a)$, which is more accurate than (20), will be used. The formula is[3]

$$B(x, a)^{-1} \cong \sum_{j=0}^{[x+a]} x^{(j)} a^{-j}. \tag{123}$$

Let

$$U_j = x^{(j)} a^{-j}. \tag{124}$$

Then, the relative error, ϵ, in calculating $B(x, a)$ is

$$\epsilon = -B(x, a) U_{[x+a]}. \tag{125}$$

A computer program employing (123) is given in the Appendix. This program combines all computations needed. It accepts m, z, S and yields r, P, EW. A test based on $U_{[x+a]}$ is made: If $U_{[x+a]} \leq 10^{-7}$, the computations are accepted and the results printed; this also reduces the time of computation by reducing the number of times a particular loop is used. If, however $U_{[x+a]} > 10^{-7}$, then the computation is considered inaccurate and the message `not accurate` is printed. This situation occurs when the offered load, m, is not large and the peakedness is near one. For example, $m = 5$, $z = 1.1$, $S = 7$ prints `not accurate`. Formula (123) is not as robust as (20) and therefore was not used in previous computations of this paper. If greater accuracy is needed in those computations, then (123) may be used in place of (20) with the required test for accuracy. Table IV gives some sample calculations.

X. SUMMARY

The mathematical methods and algorithms presented here enabled the construction of convenient computer programs that rapidly evaluate many traffic-related problems. These represent some of the main approaches and approximations used in this area of traffic theory.

Some of the routines used in the individual programs are common.

Table IV—Examples of GT/M/S computations

m	z	s	r	P	EW
12	1.1	15	0.8160	0.3444	0.1248
5	3	8	0.8750	0.5132	0.5132
2	5	5	0.8882	0.6091	1.0894
2	3	4	0.8418	0.5814	0.9186
1	4	3	0.8468	0.6392	1.3911
1	3	2	0.8505	0.7446	2.4896
0.7	4	1	0.9354	0.9354	14.4851

For this reason, a main index-driven program containing all of the programs given would shorten coding and enable the user to select one program after another as the need arises.

REFERENCES

1. B. V. Gnedenko and I. N. Kovalenko, *Introduction to Queueing Theory*, Jerusalem: S. Monson, 1968, Chapter 1.
2. R. Syski, *Introduction to Congestion Theory in Telephone Systems*, Edinburgh: Oliver and Boyd, 1959.
3. D. L. Jagerman, "Some Properties of the Erlang Loss Function," B.S.T.J., *53*, No. 3 (March 1974), pp. 525–51.
4. A. Erdelyi, *Bateman Manuscript Project*, Vol. 1, New York: McGraw-Hill, 1953.
5. R. I. Wilkinson, "Theories of Toll Traffic Engineering in the U.S.A.," B.S.T.J., *35*, No. 2 (March 1956), pp. 421–514.
6. A. A. Fredericks, "Congestion in Blocking Systems—A Simple Approximation Technique," B.S.T.J., *59*, No. 6 (July-August 1980), pp. 805–27.
7. C. Froberg, *Introduction to Numerical Analysis*, Second Edition, Reading, MA: Addison-Wesley, 1965.
8. H. Akimaru and T. Nishimura, "The Derivatives of Erlang's B Formula," Rev. Elec. Commun. Lab, *11*, No. 9-10, 1963.
9. Hardy, Littlewood, and Polya, *Inequalities*, New York: Cambridge University Press, 1934.
10. L. Y. Rapp, "Planning of Junction Networks in a Multi-Exchange Area-Part 1," Ericson Technics, *20*, 1964, pp. 77–130.
11. P. J. Burke, "The Overflow Distribution for Constant Holding Time," B.S.T.J., *50*, No. 10 (December 1971), pp. 3195–210.
12. A. E. Eckberg, unpublished work.
13. A. Kuczura, "The Interrupted Poisson Process as an Overflow Process," B.S.T.J., *52*, No. 3 (March 1973), pp. 437–48.
14. L. Takaćs, *Introduction to the Theory of Queues*, New York: Oxford, 1962.
15. V. D. Krylov, *Approximate Calculation of Integrals*, New York: MacMillan Co., 1962.
16. A. Kashper, S. M. Rocklin, and C. R. Szelag, "Effects of Day-to-Day Load Variation on Trunk Group Blocking," B.S.T.J., *61*, No. 2 (February 1982), pp. 123–35.
17. H. Heffes, "Analysis of First-Come-First-Served Queueing Systems With Peaked Inputs, B.S.T.J., *52*, No. 7 (September 1973), pp. 1215–28.
18. R. B. Cooper, *Introduction To Queueing Theory*, Second Edition, North Holland, 1981.

APPENDIX

Fortran Programs for Main Calculations in Text

Erlang Loss Function

```
          IMPLICIT REAL *8 (A-H,O-Z)
          REAL*8 MO, M1
          WRITE (6,10)
10        FORMAT (/,1X,4X,'ERLANG LOSS FUNCTION'
     2    //,1X,'NO. OF TRUNKS =')
```

```
          READ (5,20) X
20        FORMAT (F7.4)
          WRITE (6,30)
30        FORMAT ( 1X,'OFFERED LOAD =' )
          READ (5,20) A
          N = IDINT (X)
          H = X-N
          B = 1.
          DO 50 J = 1,N
          B = J/A * B + 1
50        CONTINUE
          B1 = (N+1)/A * B + 1.
          B2 = (N+2)/A * B1 + 1.
          Y = B**(1.-H)*B1**H
          Y = Y * (B1*B1/B/B2)**(H*(1.-H)/2.)
          B = 1./Y
          MO = A*B
          ZO = 1. - MO +A/(X+1.+MO-A).
          M1 = IDINT (100000.*MO+.5)/100000.
          Z1 = IDINT (100000.*ZO+.5)/100000.
          C = IDINT (100000.*B + .5)/100000.
          WRITE (6,70) X,A,B, M1,Z1
70        FORMAT (/,1X,'B(',F8.4,',',
     2    F8.4,')=' , F7.5,/,1x,
     3    'OVERFLOW =', F8.4,/,1x,
     4    'PEAKEDNESS =', F8.4,////)
          STOP
          END
```

Offered Load From Blocking

```
          IMPLICIT REAL*8(A-H,O-Z)
          WRITE (6,10)
10        FORMAT (/,1X,2,'OFFERED LOAD FROM',
     2    'BLOCKING',//1X, 'TRUNKS = ')
          READ (5,20) N
20        FORMAT (I3)
          WRITE (6,30)
30        FORMAT ( 1X, 'BLOCKING PROB. = ')
          READ (5,40) P
40        FORMAT (F4.4)
          A = N /( 1.-P) - 1.
          DO 60 K=1, 5
          B = 1.
          DO 50 J = 1, N
          B = J/A * B + 1.
50        CONTINUE
          F = 1.-P*B
          F1 = N/A-1. + 1./B
          A = A - F/F1
60        CONTINUE
          C = IDINT (10000.*A+.5)/10000.
          WRITE (6,70) C
```

```
70          FORMAT (/, 1X, 'OFFERED LOAD = ',F6.4)
            STOP
            END
```

Offered Load From Carried Load

```
            IMPLICIT REAL*8 (A-H,O-Z)
            REAL*8 L
            WRITE (6,10)
10          FORMAT (/, 1X,2X, 'OFF.LOAD FROM',
       2    'CAR. LOAD',//, 1X, 'TRUNKS = ')
            READ(5,20)N
20          FORMAT (I3)
            WRITE(6,30)
30          FORMAT( 1X, 'CAR LOAD = ')
            READ(5,40) L
40          FORMAT (F7.4)
            A = L*( 1. + L/N/ (N-L))
            DO 60 K=1,5
            B = 1.
            DO 50 J=1,N
            B = J/A*B + 1.
50          CONTINUE
            B = 1./B
            F = A*(1.-B) - L
            F1 = 1.-B - (N-A+A*B) * B
            A = A - F/F1
60          CONTINUE
            C = IDINT( 10000.*A+.5)/10000.
            WRITE (6,70) C
70          FORMAT (/, 1X, 'OFFERED LOAD = ',
       2    F8.4,////)
            STOP
            END
```

Trunks From Blocking

```
            IMPLICIT REAL*8 (A-H,O-Z)
            WRITE (6,10)
10          FORMAT (/, 1X,4X, 'TRUNKS FROM',
       2    'BLOCKING',//, 1X, 'OFFERED LOAD = ')
            READ (5,20)A
20          FORMAT (F7.4)
            WRITE(6,30)
30          FORMAT ( 1X, 'BLOCKING PROB. = ')
            READ (5,40)P
40          FORMAT (F4.4)
            X = ( 1.-P)*( 1.+A)
            DO 60 K=1,10
            N = IDINT (X)
            H = X-N
            B = 1.
```

```
      DO 50 J=1,N
      B = J/A*B+1
50    CONTINUE
      B1 = (N+1)/A*B+1.
      B2 = (N+2)/A*B1+1.
      Y = B**(1.-H)*B1**H
      Y = Y*(B1*B1/B/B2)**(H*(1.-H)/2)
      G = (X+1.)/A+1./Y
      X = X + (1.-P*Y)/DLOG(G)
60    CONTINUE
      C = IDINT (10000.*X+.5)/10000.
      WRITE (6,70) C
70    FORMAT (/,1X,'NO. OF TRUNKS = ',
     2 F8.4)
      STOP
      END
```

Equivalent Random Parameters

```
      IMPLICIT REAL*8(A-H,O-Z)
      WRITE (6,10)
10    FORMAT(/,1X,5X,'EQUIV. RAND.',
     2 'PARA.',//,1X,'LOAD = ')
      READ (5,20) M
20    FORMAT (F7.4)
      WRITE (6,30)
30    FORMAT (1X,'PEAKEDNESS = ')
      READ (5,40)Z
40    FORMAT (F7.4)
      A = M*Z+3.*Z*(Z-1.)
      X = A*(M+Z)/(M+Z-1.) - M. - 1.
      DO 70 K=1,30
      N=IDINT(X)
      H=X-N
      B=1.
      DO 50 J =1,N
      B=J/A*B+1.
50    CONTINUE
      B1=(N+1.)/A*B+1.
      B2=(N+2.)/A*B1+1.
      Y=B**(1.-H)*B1**H
      Y=Y*(B1*B1/B/B2)**(H*(1.-H)/2.)
      B=1./Y
      G=(X+1.)/A + B
      R=A*B-M
      D=B*(X+M+1.-A-(X+M+1.)*DLOG(G))
      A=A-R/D
      X=A*(M+Z)/(M+Z-1.)-M-1
70    CONTINUE
      X1=IDINT(10000.*X+.5)/10000.
      A1=IDINT(10000.*A+.5)/10000.
      WRITE (6,80)X1,A1
80    FORMAT (/,1X,'EQUIV.TRUNKS = ',
```

```
      2         F8.4,/,1X,'EQUIV. LOAD = ', F8.4)
                STOP
                END
```

Blocking E.R.M.

```
                IMPLICIT REAL*8 (A-H,O-Z)
                REAL*8 M2, M2, M
                WRITE (6,10)
 10             FORMAT (/,1X,6X, 'BLOCKING E.R.M.',
      2         //,1X, 'TRUNKS = ')
                READ (5,20) C
 20             FORMAT (F7.4)
                WRITE (6,30)
 30             FORMAT( 1X,'OFFERED LOAD = ')
                READ (5,20) M
                WRITE (6,40)
 40             FORMAT ( 1X, 'PEAKEDNESS = ')
                READ (5,50) Z
 50             FORMAT (F7.4)
                A= M*Z+3.*Z* (Z-1)
                X= A* (M+Z)/ (M+Z-1.) - M-1.
                DO 80 K=1,25
                CALL ERLNG (A,X,B)
                G = (X+1.)/A + B
                A = A - (A*B-M)/B/(X+M+1.-A-
      2         (X+M+1.)* DLOG (G))
                X = A* (M+Z)/(M+Z-1.) - M-1.
 80             CONTINUE
                X = X +C
                CALL ERLNG2 (A,X,B)
                P1 = B
                MO = A*B
                ZO = 1.-M0+A/(X+1.+MO-A)
                M2 = IDINT ( 10000.*MO+.5)/10000.
                Z2 = IDINT ( 10000.*ZO+.5)/10000.
                X=X-C
                CALL ERLNG(A,X,B)
                P = P1/B
                P2 = IDINT ( 10000.*P+.5)/10000.
                WRITE (6,100) P2, M2, Z2
 100            FORMAT (/,1X,'BLOCKING E.R.M. = ',
      2         F6.4,/,1X, 'OVERFLOW TRAFFIC =',
      3         F8.4,/,1X, 'OVERFLOW PEAKEDNESS =',
      4         F8.4)
                STOP
                END
```

Three-Moment Match

```
                IMPLICIT REAL*8 (A -H, O-Z)
                REAL*8 L, L1, M, M1
                DIMENSION D(3)
```

```
      WRITE (6,10)
10    FORMAT (/,1x,5x, 'THREE MOMENT',
    2 'MATCH',//,1X, 'EQUIV. TRUCKS = ',)
      READ (5,20) X
20    FORMAT (F7.4)
      WRITE (6,30)
30    FORMAT ( 1X, 'EQUIV. LOAD = ')
      READ (5,20) A
      B=1.
      N=IDINT (X)
      H=X-N
      DO 50 J=1,N
      B=J/A*B+1.
50    CONTINUE
      B1 = (N+1)/A*B+1.
      B2 = (N+2.)/A*B1+1.
      Y = B**(1.-H)*B1**H
      Y = Y*(B1*B1/B/B2)*(H*(1.-H)/2.)
      D(1) = 1./Y
      B3=(X+1.)/A*Y + 1.
      D(2)=Y/A/(B3-Y)
      B4=(X+2.)/A*B3 + 1.
      D(3)=2./A/A*Y/D(2)/(B4-2*B3+Y)
      L= D(3)*(D(2)-D(1))-D(1)*(D(3)-D(2))
      L= L/(2.*D(2)-D(1)-D(3))*A
      W= D(1)/L*(L-A* D(2))/(D(2)-D(1))
      G= W/A*(L-A*D(11))/D(1)
      L1= IDINT(10000.*L+.5)/10000.
      W1= IDINT(10000.*W+.5)/10000.
      G1= IDINT(10000.*G+.5)/10000.
      M= L*W/(G+W)
      M1= IDINT(10000.*M+.5)/10000.
      Z= 1.+L*G/(G+W)/(1.+G+W)
      Z1= IDINT (10000.*Z+.5)/10000.
      WRITE (6,80) L1, W1, G1, M1, Z1
80    FORMAT (/,1X, 'L=',F7.4,/,1X,
    2 'W =',F7.4,/, 1X,'G = ',F7.4,/,1X,
    3 'M =',F7.4,/, 1X,'Z = ',F7.4)
      STOP
      END
```

Blocking Day-to-Day Time

```
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 M,MB
      WRITE (6,10)
10    FORMAT (/, 1X,4X,'BLOCKING-DAY TO',
    2   'DAY TIME',//,1X,'TRUNKS = ')
      READ (5,20) X
20    FORMAT (F7.4)
      WRITE (6,30)
30    FORMAT ( 1X, 'MEAN LOAD = ')
      READ (5,40) M
```

```
40        FORMAT (F7.4)
          WRITE (6,50) V
50        FORMAT ( 1X, 'VARIANCE = ')
          READ (5,60)V
60        FORMAT (F7.4)
          W1 = ( 1-DSQRT(V/(M*M+V)))/2.
          A = (M*M+V+DSQRT (V*(M*M+V)))/M
          A1 = A
          CALL ERLNG (A,X,B)
          MB = W1*B
          W2 = 1. - W1
          A = (M*M+V-DSQRT(V*(M*M+V)))/M
          A2 = A
          CALL ERLNG (A,X,B)
          MB = MB + W2*B
          C = IDINT ( 10000.*MB+.5)/10000.
          WRITE (6,90) C
90        FORMAT (/, 1X, 'MEAN BLOCKING = ',
     2    F6.4, ////)
          STOP
          END
```

Blocking Day-to-Day Call

```
          IMPLICIT REAL*8 (A-H,O-Z)
          REAL*8 M,MB
          WRITE (6,10)
10        FORMAT (/, 1X,4X, 'BLOCKING-DAY TO',
     2      'DAY CALL',//, 1X, 'TRUNKS = ')
          READ (5,20)X
20        FORMAT (F7.4)
          WRITE (6,30)
30        FORMAT ( 1X, 'MEAN LOAD =')
          READ (5,40)M
40        FORMAT (F7.4)
          WRITE (6,50)
50        FORMAT ( 1X, 'VARIANCE = ')
          READ (5,60)
60        FORMAT (F7.4)
          W1 = ( 1.-DSQRT(V/(M*M+V)))/2.
          A = (M*M+V+DSQRT (V*(M*M+V)))/M
          A1 = A
          CALL ERLNG (A,X,B)
          MG = W1*A*B
          W2 = 1.-W1
          A = (M*M+V-DSQRT (V*(M*M+V)))/M
          A2 =A
          CALL ERLNG (A,X,B)
          MB = MB+W2*A*B
          C = IDINT ( 10000.*MB/M+.5)/10000.
          WRITE (6,90)C
90        FORMAT (/, 1X, 'MEAN BLOCKING = ',
```

```
      2      F6.4,///)
             STOP
             END

             SUBROUTINE ERLNG (A,X,B)
             IMPLICIT REAL*8(A-H, O-Z)
             H=X-N
             B = 1.
             DO 30 J = 1,N
             B = J/A*B+1.
  30         CONTINUE
             B1 = (N+1.)/A*B+1.
             B2 = (N+2.)/A*B1+1.
             Y = B**(1.-H)*B1**H
             Y = Y*(B1*B1/B/B2)**(H*(1.-H)/2.)
             B = 1./Y
             RETURN
             END
```

Waiting Time in GT/M/S Queue

```
             IMPLICIT REAL*8(A-H,O-Z)
             REAL*8 L,M,N1
             INTEGER S
             DIMENSION D(3)
             WRITE (6,10)
  10         FORMAT (/,1x,'WAITING TIME IN GT/M/S',
      2        'QUEUE',//,1x,'OFFERED LOAD =')
             READ(5,20) M
  20         FORMAT(F7.4)
             WRITE(6,30)
  30         FORMAT(1x,'PEAKEDNESS = ')
             READ(5,40)Z
  40         FORMAT (F7.4)
             WRITE(6,50)
  50         FORMAT(1X,'NO. OF SERVERS = ')
             READ(5,60)S
  60         FORMAT(I3)
             A = M*Z+3.*Z*(Z-1.)
             X = A*(M+Z)/(M+Z-1.)-M-1.
             DO 100 K=1,30
             N = X+A
             U2 = 1.
             Y = 1.
             DO 80 J = 1,N
             U2 = U2*(X-J+1)/A
             Y = Y+U2
             IF (DABS(U2).LE.1E-7) GO TO 90
  80         CONTINUE
  90         B = 1./Y
             G = (X+1.)/A+B
             N1 = A*B-M
             D1 = B*(X+M+1.-A-(X+M+1.)*DLOG(G).
```

```
                A = A-N1/D1
                X = A*(M+Z)/(M+Z-1.)-M-1.
100             CONTINUE
                IF (DABS(U2).GT.1E-7) GO TO 200
                D(1) = B
                B3 = (X+1.)/A*Y+1.
                D(2) = Y/A/(B3-Y)
                B4 = (X+2.)/A*B3+1.
                D(3) = 2./A/A*Y/D(2)/(B4-2.*B3+Y)
                L = D(3)*(D(2)-D(1))-D(1)*(D(3)-D(2))
                L = L/(2.*D(2)-D(1)-D(3))*A
                W = D(11)/L*(L-A*D(2))/(D(2)-D(1))
                G = W/A*(L-A*D(1))/D(1)
                V = (L+G+W)*S-S*S
                V1 = (G+W)*S-L*W
                D2 = (DSQRT(V*V+4.*V1*S*S)-V)
          2     /2./S/S
                C = 1.
                E = 1.
                U1 = 1./D2
                DO 150 J = 1,S
                E = E*(S-J+1.)/J
                C = C*(1./F(J,L,W,G) - 1.)
                U = E/C/F(J,L,W,G)*(S*F(J,L,W,G)
          2     -J)/(S*D2-J)
                U1 = U1 + U
150             CONTINUE
                R = 1.-D2
                P = 1./U1/D2
                W1 = P/D2/S
                WRITE(6,170)R,P,W1
170             FORMAT(/,1x,'ROOT = ',F8.4,/,
          2       1X,'P(W>0) = ',F6.4,/,
          3       IX,'MEAN W = ',F8.4)
                STOP
200             WRITE (6,210)
210             FORMAT (/,1X,'NOT ACCURATE')
                STOP
                END
                REAL FUNCTION F*8(J,L,W,G)
                IMPLICIT REAL*8(A-H,O-Z)
                REAL*8 L
                F = (J*J+(G+W)*J)/(J*J+
          2       (L+G+W)*J+L*W)
                RETURN
                END
```

AUTHOR

David L. Jagerman, B.E.E. (Electrical Engineering), 1949, Cooper Union; M.S., and Ph.D. (Mathematics), 1954 and 1962, respectively, New York University; AT&T Bell Laboratories, 1963—. Mr. Jagerman has been engaged in mathematical research on quadrature, interpolation, and approximation

theory, especially related to the theory of widths and metrical entropy, with application to the storage and transmission of information. For the past several years, he has worked on the theory of difference equations and queueing, especially with reference to traffic theory and computers. He is currently preparing a text on difference equations with application to stochastic models.