Q^2: A New Performance Analysis Tool Exploiting Numerical Transform Inversion

Gagan L. Choudhury

AT&T Bell Laboratories
Holmdel, NJ 07733-3030

Ward Whitt

AT&T Bell Laboratories
Murray Hill, NJ 07974-0636

Proceedings of the third International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (*MASCOTS '95*), Durham, NC, January 1995, pages 411–415.

Abstract

We describe a new tool that we are developing to help analyze the performance of emerging telecommunication systems. These emerging systems include ATM, broadband, intelligent and wireless networks, each of which will support a wide variety of services and media. Our tool is distinguished from previous tools by exploiting numerical transform inversion. In this paper we describe three modules in the tool. These modules provide algorithms for obtaining the exact analytic solutions to: (i) resource-sharing models, (ii) BMAP/G/1 queueing models and (iii) polling models. Since the tool is based on analytical methods instead of simulation, it is relatively fast. The tool has a window-based menu-driven interface and provides both graphical and numerical output.

1 Introduction

In this paper we describe a new performance analysis tool that we are developing, called Q^2, which exploits recent progress in numerical transform inversion. The tool has a user-friendly interface with window-based menu-driven input and graphical as well as numerical output. Since the algorithms are based on analytical solutions instead of simulation, the running times are relatively short. Many substantial problems can be analyzed in fractions of a second or seconds, when simulations would take hours or days.

As usual, the computational engine of the performance analysis tool is being kept mostly transparent to the user. Hence, the tool can include a wide variety of different algorithms without excessively burdening the user. However, for developing, maintaining, understanding and fine-tuning the tool, it is significant that all the initial modules are based on a single technology: numerical transform inversion. It is remarkable that this one approach has so many applications to performance analysis models. Indeed, it is surprising that it has not previously been given much more attention.

Numerical transform inversion has several advantages over other computational approaches for performance analysis models. For many stochastic models of interest, transforms of key quantities either are already available or are relatively easy to obtain. Moreover, transform inversion tends to place fewer restrictions on the distributions appearing in the model. For example, in the GI/G/1 queue the interarrival-time and service-time distributions need not be phase-type or even have rational Laplace transforms [1]. Numerical inversion also routinely produces tail probabilities instead of just means. It is now widely recognized that tail probabilities usually are more informative in performance analysis than means. Moreover, simulation is usually quite effective for computing means, but simulation has difficulty computing small tail probabilities.

Hence, the essential tool behind the tool is numerical transform inversion. For this purpose, we exploit the Fourier-series method, drawing upon our previous work in [2], [7], [8]. We are able to invert both one-dimensional and multi-dimensional transforms, where these transforms can be generating functions (or z-transforms), Laplace transforms, Fourier transforms (or characteristic functions) or combinations of these.

This paper only provides a brief overview, but it indicates where additional information can be found. This paper can be regarded as a sequel to our overview of transform inversion applications in [9]. Here we discuss three different models for which we have successfully applied numerical inversion: (i) resource-sharing models, (ii) the BMAP/G/1 queue and (iii) polling models. These models are the first three modules in Q^2. Their relevance for performance analysis is well known. All three models lead to relatively complex structured Markov chains. The special structure enables us to obtain relatively compact representations of the steady-state distributions via transforms. In the last two cases we have transforms of the transient distributions as well. For many cases of the three models, numerical transform inversion appears to be the only available technique for computing distributions of interest.

2 Resource-sharing models

The first module of Q^2 is for solving resource-sharing models, drawing on our work with K. K. Leung in [3]–[6]. Resource-sharing models (also known as loss networks) are multi-dimensional generalizations of the classical Erlang loss model. In a resource-sharing model there are multiple *resources*, each containing multiple resource *units* which provide service to multiple *customers*. Each customer is a source of a stream of *requests*. Each customer request requires a number of units from each resource, which may be zero, one or greater than one, and may be different on different resources and different for different customers. If all requirements can be met upon arrival of a new request, then the new request is admitted, and all required resource units are held throughout the request holding time. Otherwise, the request is blocked and lost. The primary measures of performance are the *request blocking probabilities* of the different customers.

In a circuit-switched telecommunications network, the resources may be links, and the resource units may be circuits on these links, while the customers may be associated with different services and the requests may be calls. In an ATM network, the resources may be switches and other network facilities, the resource capacity (units) may be the bandwidth available at these network facilities, while the customers may be prospective users of the network and the customer requests may be required "effective bandwidths" associated with bursts within an established connection. Thus this is a candidate model to address the well known ATM call-admission-control problem. Both applications are intended for a broad range of services, so that it is important that the model includes customers with different characteristics. In particular, it is important to allow some customer requests to use multiple resource units.

The standard resource-sharing model has a *complete-sharing* (CS) policy, in which requests are admitted whenever all the required resource units are free. However, we consider more general resource-sharing policies involving extra linear constraints. We pay particular attention to the case in which *upper-limit* (UL) and *guaranteed minimum* (GM) bounds are assigned to each customer. A UL bound limits the number of requests from that customer that can be in service. A GM bound guarantees that there is always space for a specified number of active requests from that customer. A set of GM bounds is equivalent to an upper limit on the resource units used by each subset of the classes. The UL and GM bounds are equivalent for two classes, but not for more than two classes. We focus on *combined UL and GM bounds* (which cannot be reduced to either one alone). The UL and GM bounds are very useful for providing protection against overloads and for providing different grades of service to different customers.

In the standard resource-sharing model, the request arrival processes are independent Poisson processes and the request holding times are independent random variables with a general distribution having finite mean. The key description of each customer's request stream is the offered load, which is the product of the arrival rate and the mean holding time. It is well known that this resource-sharing model has a *product-form steady-state distribution*, i.e., if $\mathbf{n} \equiv (n_1, \ldots, n_r)$, where n_j is the number of customer-j requests in service, then the *steady-state probability mass function* is

$$\pi(\mathbf{n}) = g(\mathbf{K})^{-1} f(\mathbf{n}), \text{with } f(\mathbf{n}) = \Pi_{j=1}^{r} f_j(n_j) \quad (2.1)$$

and $f_j(n_j) = \exp(-\rho_j)\rho_j^{n_j}/n_j!$ (the Poisson distribution) where ρ_j is the offered load for class j, and the *normalization constant* (or partition function) $g(\mathbf{K})$ in (2.1) is the sum of $f(\mathbf{n})$ over all allowable states. The set of allowable states depends on the sharing policy. With the complete-sharing policy, the quantity $\mathbf{K} \equiv (K_1, \ldots, K_p)$ in $g(\mathbf{K})$ is the vector of resource capacities; otherwise it is more complicated. Moreover, it is known that the blocking probabilities and other steady-state performance measures of interest have simple expressions in terms of the normalization constants appearing in the steady-state distribution.

In this module we use our new algorithm for calculating the blocking probabilities based on *numerically inverting the generating function (or z-transform) of the normalization constant* [3]. If the vector $\mathbf{K}$ in the normalization constant $g(\mathbf{K})$ is p-dimensional, then the generating function is the following p-dimensional function of complex variables $\mathbf{z} \equiv (z_1, \ldots, z_p)$:

$$G(\mathbf{z}) = \sum_{i_1=1}^{\infty} \cdots \sum_{i_p=1}^{\infty} g(\mathbf{K}) z_1^{K_1} \ldots z_p^{K_p} \ . \qquad (2.2)$$

For our approach, it is significant that the generating function has a remarkably simple form. For example, with the complete-sharing policy, the generating function is

$$G(\mathbf{z}) = \Pi_{i=1}^{p} (1 - z_i)^{-1} \exp(\sum_{j=1}^{r} \rho_j \Pi_{i=1}^{p} z_i^{a_{ij}}) \ , \qquad (2.3)$$

where p is the number of resources, r is the number of customers and a_{ij} is the number of units of resource i required by each customer-j request. With the combined UL and GM bounds, the generating function is more complicated but still tractable.

We also provide variants of the algorithm which have been developed for state-dependent arrival and service rates [4] and batch arrivals [5]. State-dependent arrival rates and batch arrivals are important to represent sources of customer requests that are more or less bursty (variable) than a Poisson process. State-dependent service rates permit us to consider buffered as well as unbuffered models, such as the single-server variant considered by Kamoun and Kleinrock [13] to analyze a node of a store-and-forward computer network.

Recursive algorithms, such as the Kaufman [14] – Roberts [21] algorithm, have previously been developed for many resource-sharing models, but not in the generality above. Moreover, for those models for which recursive algorithms have been developed, our inversion algorithm is computationally superior for large models. The numerical inversion algorithm has a number of computational advantages. First, large finite sums can be efficiently computed through judicious *truncation* or through *acceleration methods.* Second, for large models with a high-dimensional generating function, it is often possible to *reduce the effective dimension* by inverting the variables in a good order. For example, this dimension reduction enables us to solve models with UL and GM bounds nearly as quickly as the standard model with the CS sharing policy. It is also possible to reduce the computations by exploiting *multiplicities*, i.e., multiple classes with identical parameters. We can make our models much larger by increasing multiplicities at negligible computational cost.

Our algorithm exploits the Fourier-series method for inverting generating functions, as in [2], [8]. Since the normalization constants can grow rapidly in the resource capacities, an important ingredient in our inversion algorithm is an effective scaling procedure [3], [4]. We demonstrate the effectiveness of the overall inversion algorithm in [3], [4], [5] by solving some challenging numerical examples.

Even though the inversion algorithm is remarkably effective, very large networks without special structure are well beyond the capabilities of the inversion algorithm. To approximately solve very large models, following [6], the module uses reduced-load fixed-point approximations, exploiting the inversion algorithm for solving single resources or subnetworks.

3 The BMAP/G/1 queue

The second module of Q^2 computes steady-state and transient performance measures for the BMAP/G/1 queue, drawing on our work with D. M. Lucantoni in [7]–[11], [17]. The BMAP/G/1 queue is a single-server queue with unlimited waiting space, the first-come first-served service discipline, independent and identically distributed service times with a general distribution, and a *batch Markovian arrival process* (BMAP), which is independent of the service times. The most significant feature is the BMAP. It is a very general arrival process, permitting a rich variety of models [16]. The BMAP is an alternative representation for the *versatile Markovian point process* or *Neuts process* [19], [20].

The BMAP can be constructed by considering a two-dimensional Markov process $\{[N(t), J(t)] : t \geq 0\}$ on the state space $\{i, j) : i \geq 0, 1 \leq j \leq m\}$ with an *infinitesimal generator* Q having the structure

$$Q = \begin{pmatrix} D_0 & D_1 & D_2 & D_3 & \ldots \\ & D_0 & D_1 & D_2 & \ldots \\ & & D_0 & D_1 & \ldots \\ & & & D_0 & \ldots \end{pmatrix} \ , \qquad (3.1)$$

where $D_k, k \geq 0$, are $m \times m$ matrices; D_0 has negative diagonal elements and nonnegative off-diagonal elements; $D_k, k \geq 1$, are nonnegative; and $D \equiv \sum_{k=0}^{\infty} D_k$ is an irreducible infinitesimal generator. We assume that $D \neq D_0$, so that arrivals do occur. The variable $N(t)$ counts the number of arrivals in the interval $(0, t]$, and the variable $J(t)$ represents an auxiliary *state.* Transitions from (i, j) to $(i + k, l), k \geq 0$, $1 \leq j, l \leq m$, correspond to batch arrivals of size k

along with a change of state from j to l, and these occur with intensity $(D_k)_{jl}$.

Note that it is possible to have any of: (i) arrivals without change of auxiliary state, (ii) arrivals with change of auxiliary state, and (iii) change of auxiliary state without arrivals. A familiar special case of a BMAP is the *Markov modulated Poisson process* (MMPP) having an m-dimensional (diagonal) rate matrix Λ. (The environment is governed by a Markov chain with generator M. When the chain is in state j, arrivals occur according to a Poisson process with rate λ_j.) An MMPP is a BMAP with $D_0 = M - \Lambda, D_1 = \Lambda$ and $D_k = 0$ for $k \geq 2$.

Since a superposition of independent BMAPs is again a BMAP, the BMAP is very useful to study queues with superposition arrival processes, which in turn are useful to study the phenomenon of statistical multiplexing in communication networks. For ATM networks there is great interest in small cell loss probabilities associated with large numbers of independent sources. We have used our algorithm for the BMAP/G/1 queue to gain insight into the ATM problem by computing small tail probabilities in a BMAP/G/1 queue with an arrival process that is a superposition of 60 two-state MMPPs [9], [11].

The BMAP/G/1 queue is a special case of a structured Markov chain of M/G/1 type [19]. Within the class of Markov chains of M/G/1 type, the BMAP/G/1 queue has a special place, because virtually all its performance measures can be expressed as matrix generalizations of the corresponding performance measures in the ordinary M/G/1 queue [16]. However, just as for the ordinary M/G/1 queue, many of the desired probability distributions in the BMAP/G/1 queue are only available (except in special cases) in the form of transforms.

Matrix-analytic theory has provided expressions for the transforms of quantities of interest [15], [16], [19], [20], in the BMAP/G/1 queue and M/G/1-type Markov chains. However, in contrast to the situation for GI/M/1-type Markov chains [18], there has remained a need for effective algorithms for the BMAP/G/1 queue and M/G/1-type Markov chains. It appears that numerical transform inversion is a natural tool to fill this gap.

To illustrate, we consider the queue-length process at departure epochs in the BMAP/G/1 queue, which is an M/G/1-type Markov chain. Let x_i represent the steady-state queue-length vector, whose j^{th} element is the probability that the queue length is i and the arrival process is in state j right after a departure in steady state. Its generating function is

$$X(z) \equiv \sum_{i=0}^{\infty} x_i z^i = -x_0 D_0^{-1} D(z) A(z)[zI - A(z)]^{-1}, \tag{3.2}$$

where D_0 is from (3.1), $D(z)$ is the matrix generating function $D(z) \equiv \sum_{k=0}^{\infty} D_k z^k$ and $A(z) \equiv \tilde{A}(z,0)$, where $\tilde{A}(z,s)$ is the two-dimensional matrix transform

$$\tilde{A}(z,s) = \int_0^{\infty} e^{-x(sI - D(z))} dH(x) \equiv \hat{h}(sI - D(z)), \tag{3.3}$$

with H being the service-time cumulative distribution function and $\hat{h}$ its Laplace-Stieltjes transform. It is known [15] that x_0 in (3.2) is computable in terms of the model input, so that the most challenging part is (3.3). However, as shown in [17], when H has a rational Laplace transform, it is not difficult to compute $\tilde{A}(z,s)$ for any pair of complex numbers (z,s). In addition, the running time is quite insensitive to the degree of the polynomials. Hence, we are able to calculate x_i by numerically inverting (3.2).

To make the tool easy to use, in addition to allowing the full generality of the model, we consider special cases of the model that can be specified by relatively few parameters. For example, we consider Erlang (E_k) and hyperexponential (H_k) service-time distributions. An H_2 distribution can be represented as a mixture of two exponential distributions or as the interarrival time in an interrupted Poisson process, which is a special case of a two-state MMPP in which one rate is 0. We allow the three H_2 parameters to be specified by the natural parameter triples associated with these two representations or by the first three moments.

For the arrival process, we allow a class of general BMAPs by having $D_k = 0$ for $k > m$ for some m and having the user specify m and D_k, $0 \leq k \leq m$. We also allow superpositions of two special BMAPs. The first special class contains E_k and H_k renewal processes. The second special class contains MMPPs. The user specifies the number of basic processes (renewal processes or MMPPs), the multiplicity of each in the superposition, and the parameters for each. The tool then constructs the BMAP representation for the overall superposition process. At present, our algorithm can solve models in which the final BMAP has up to about 100 states.

In addition to calculating probability distributions, we also use numerical inversion to calculate any number of their moments and their asymptotic parameters [7]. As with the normalization constants in the resource-sharing models, it is important to have an effective scaling procedure to compute high-order moments. As shown in [7], the high-order moments can in turn be used to compute the asymptotic parameters. For example, it often happens that

$$x_{ij} \sim \alpha_j \sigma^i \text{ as } i \to \infty, \tag{3.4}$$

where $a_i \sim b_i$ as $i \to \infty$ means that the ratio converges to 1. The program computes the parameters α_j and σ in (3.4). The approximation $x_{ij} \approx \alpha_j \sigma^i$ is a convenient compact representation that is often adequately accurate for many applications [10], [11]. We also have refined three-term approximations that match moments and asymptotics, which are even more accurate.

4 Polling models

The third module of Q^2 calculates steady-state and transient performance measures for polling models, drawing on [12]. In our polling models, independent Poisson arrivals come to different queues, where they are served by a single server. The server may serve the queues in a *cyclic order* or according to a more general *polling table*. We allow either gated or exhaustive service at each queue, with different policies allowed at different queues. With the *gated policy*, the server serves all customers found at the queue when it first arrives there, but none of the customers that arrive later. With the *exhaustive policy*, the server keeps working until the queue is empty, serving new customers who arrive while the server is busy serving earlier arrivals at that queue. In [12] we show that the inversion approach applies to many other polling models as well.

The time required for the server to move from one queue to another is called a *switchover time*. During a switchover time no service is performed. We treat both the case of *zero switchover times* and *nonzero switchover times*. We assume that the switchover times and service times are mutually independent random variables with general distributions that depend on the queue. As in the previous BMAP/G/1 module, the user can specify these distributions by selecting the general form such as E_k and H_k, and then providing the required parameters.

As can be seen from Takagi [22], [23], multidimensional transforms of performance measures of interest have been derived for these (and related) polling models. These transform expressions have been successfully exploited to derive means and sometimes second moments, but until [12] they evidently had not been used to calculate the distributions themselves, higher moments or asymptotic parameters.

In [12] we show that these polling transforms often can be quite easily computed and inverted numerically to calculate distributions, all moments and asymptotic parameters. We develop a new efficient recursive algorithm for computing transform values. Our operation count for computing moments and distributions is $O(N^\alpha)$ for one queue and is $O(N^{1+\alpha})$ for all queues of an N-queue system, where α is typically in the range 0.6 to 0.8. It appears that our algorithm is faster than other available algorithms for mean waiting time and queue lengths. For example, in [12] we treat an asymmetric 1000-queue system and compute the mean waiting time in less than 5 seconds, and several moments and tail probability values in a few minutes, using a SUN SPARC-2 workstation.

Computing full distributions instead of only means is important because in many performance analysis applications we really want to know high percentiles, such as the 95th or 99th. In emerging high-speed communication networks there is even great interest in very small tail probabilities, such as 10^{-9}, in order to provide an appropriate quality of service. Hence, the ability to compute full distributions should significantly increase the usefulness of polling models.

We compute all moments by numerical inversion, so that our method is the same for the hundredth moment as it is for the first moment. By contrast, previous results for higher moments have been via analytical differentiation of the transform, which leads to cumbersome expressions. So far, analytical differentiation has only provided results for the first two moments, but we can easily compute even the hundredth moment.

Acknowledgment.

We thank Joseph Abate, Kin K. Leung, David M. Lucantoni and Leon Wang for their significant contributions to this work. We also thank David J. Houck and Patricia E. Wirth for their encouragement.

References

[1] J. Abate, G. L. Choudhury and W. Whitt, “Waiting-Time Tail Probabilities in Queues with Long-Tail Service-Time Distributions,” *Queueing Systems*, vol. 16, pp. 311-338, 1994.

[2] J. Abate and W. Whitt, “The Fourier-Series Method for Inverting Transforms of Probability Distributions,” *Queueing Systems*, vol. 10, pp. 5–88, 1992.

[3] G. L. Choudhury, K. K. Leung and W. Whitt, “An Algorithm to Compute Blocking Probabil-

ities in Multi-Rate Multi-Class Multi-Resource Loss Models," *Adv. Appl. Prob.*, to appear.

[4] G. L. Choudhury, K. K. Leung and W. Whitt, "An Inversion Algorithm to Compute Blocking Probabilities in Loss Networks with State-Dependent Rates," submitted.

[5] G. L. Choudhury, K. K. Leung and W. Whitt, "Resource-Sharing Models with State-Dependent Arrivals of Batches," submitted.

[6] G. L. Choudhury, K. K. Leung and W. Whitt, "Improved Reduced-Load Fixed-Point Approximations for Large Loss Networks," in preparation.

[7] G. L. Choudhury and D. M. Lucantoni, "Numerical Computation of Moments of a Probability Distribution from its Transform," *Oper. Res.*, to appear.

[8] G. L. Choudhury, D. M. Lucantoni and W. Whitt, "Multidimensional Transform Inversion With Applications to the Transient M/G/1 queue," *Ann Appl. Prob.*, vol. 4, pp. 719–740, 1994.

[9] G. L. Choudhury, D. M. Lucantoni and W. Whitt, "Numerical Transform Inversion to Analyze Teletraffic Models," *Proceedings ITC 14*, J. Labetoulle and J. W. Roberts (eds.), Elsevier, Amsterdam, pp. 1043–1052, 1994.

[10] G. L. Choudhury, D. M. Lucantoni and W. Whitt, "On the Effectiveness of Effective Bandwidths for Admission Control in ATM Networks," *Proceedings ITC 14,* J. Labetoulle and J. W. Roberts, eds, Elsevier, Amsterdam, pp. 411–420, 1994.

[11] G. L. Choudhury, D. M. Lucantoni and W. Whitt, "Squeezing the Most Out of ATM," *IEEE Trans. Commun.*, to appear.

[12] G. L. Choudhury and W. Whitt, "Computing Transient and Steady-State Distributions in Polling Models by Numerical Transform Inversion," submitted.

[13] F. Kamoun and L. Kleinrock, "Analysis of Shared Finite Storage in a Computer Network Node Environment Under General Traffic Conditions," *IEEE Trans. Commun.*, vol. COM-28, pp. 992–1003, 1980.

[14] J. S. Kaufman, "Blocking in a Shared Resource Environment, *IEEE Trans. Commun.* vol. COM-29, pp. 1474–1481, 1981.

[15] D. M. Lucantoni, "New Results for the Single Server Queue with a Batch Markovian Arrival Process," *Stoch. Models*, vol. 7, pp. 1–46, 1991.

[16] D. M. Lucantoni, "The BMAP/G/1 Queue: A Tutorial, "*Models and Techniques for Performance Evaluation of Computer and Communication Systems*, L. Donatiello and R. Nelson, eds., Springer, New York, pp. 330–358, 1993.

[17] D. M. Lucantoni, G. L. Choudhury and W. Whitt, "The Transient BMAP/G/1 Queue," *Stoch. Models*, vol. 10, pp. 145–182, 1994.

[18] M. F. Neuts, *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach*, The Johns Hopkins University Press, Baltimore, 1981.

[19] M. F. Neuts, *Structured Stochastic Matrices of M/G/1 Type and Their Applications*, Marcel Dekker, Yew York, 1989.

[20] V. Ramaswami, "The N/G/1 Queue and its Detailed Analysis," *Adv. Appl. Prob.*, vol. 12, pp. 222–261, 1980.

[21] J. W. Roberts, "A Service System with Heterogeneous User Requirements," *Perf. of Data Commun. Systems and Their Applications*, G. Pujolle, ed., North-Holland, Amsterdam, pp. 423–431, 1981.

[22] H. Takagi, *Analysis of Polling Systems*, MIT Press, 1986.

[23] H. Takagi, "Queueing Analysis of Polling Models: An Update," *Stochastic Analysis of Computer and Communication Systems*, H. Takagi, ed., Elsevier, Amsterdam, pp. 267–318, 1990.