Modeling Multiple IP Traffic Streams With Rate Limits

Daniel P. Heyman and David Lucantoni, *Senior Member, IEEE*

***Abstract*—We start with the premise, and provide evidence that it is valid, that a Markov-modulated Poisson process (MMPP) is a good model for Internet traffic at the packet/byte level. We present an algorithm to estimate the parameters and size of a discrete MMPP (D-MMPP) from a data trace. This algorithm requires only two passes through the data. In tandem-network queueing models, the input to a downstream queue is the output from an upstream queue, so the arrival rate is limited by the rate of the upstream queue. We show how to modify the MMPP describing the arrivals to the upstream queue to approximate this effect. To extend this idea to networks that are not tandem, we show how to approximate the superposition of MMPPs without encountering the state-space explosion that occurs in exact computations.**

Numerical examples that demonstrate the accuracy of these methods are given. We also present a method to convert our estimated D-MMPP to a continuous-time MMPP, which is used as the arrival process in a matrix-analytic queueing model.

***Index Terms*—Hidden Markov model, Markov-modulated Poisson process (MMPP), matrix-analytic queueing model, superposition, tandem queues.**

I. Introduction

TWO OF THE many applications of queueing models in communications networks are sizing links in transport networks and buffers in routers. A fundamental part of a queueing model is the arrival process. A Markov-modulated Poisson process (MMPP) is an attractive model for describing backbone packet traffic. This paper describes a computationally simple method of fitting an MMPP to trace data. Suppose several traffic streams, each described by an MMPP, are going to be combined. This occurs when access links are combined at a provider edge router and when several links are replaced by a single higher-speed link. This paper provides an algorithm for combining the MMPPs for the original streams into a tractable MMPP that describes the superposition of these streams.

Once it was recognized that (at the packet or byte level) IP traffic is not described well by a Poisson process, it was natural to try and describe it by an MMPP (see, for example, [1]–[3]). The MMPP has enough flexibility to describe a wide variety of data, and its physical interpretation seems to describe rate fluctuations in many situations. For example, there are several methods of fitting an MMPP to capture the behavior over several time scales (see, e.g., [4] and [5]). Therefore, the self-similar properties described in [6] can be matched over the time scales of interest. Moreover, the time-varying Poisson properties of the MMPP is consistent with the statistical findings of [7] for data collected on highly utilized links. In addition, there is a wide body of knowledge and software available to make computable performance models [8], [9]. A challenge in describing a time series by an MMPP is to estimate the parameters of the model accurately while keeping the number of states small enough to make the performance models tractable. This paper addresses that challenge by proposing an algorithm to estimate MMPP parameters from count data.

An additional challenge is to limit state-space explosion in the description of a superposition of MMPPs. The superposition of two MMPPs is an MMPP whose transition matrix is the Kronecker sum of the transition matrices of the constituent MMPPs. The dimension of the superposition of k MMPPs with dimensions $n_1, n_2, \ldots, n_k$ is the product of these dimensions, so performance models where the arrival process is the superposition of MMPPs become intractable for k as small as two when the original MMPPs have about 20 states. We use the idea behind the MMPP estimation algorithm to approximate the superposition MMPP. We find that the number of states grows slowly with the number of MMPPs superposed.

A third contribution of this paper is motivated by the following scenario. Users generate packet arrivals in the form of an MMPP. The arrivals pass through a constant-rate device, such as an access line (so many bits per second) or a router (so many packets per second). When the user rate exceeds the rate of the device, assume that the excess is either buffered or retransmitted after a short delay. Then the output from the device is a modification of the original MMPP with a peak rate equal to the rate of the device. We present a method to capture this effect and develop equations to describe this rate-limited process.

With the discovery that much Internet traffic exhibits long-range dependence (LRD), one might wonder why we propose to use an MMPP that, by definition, is short-range dependent (SRD), as a model of IP traffic. As discussed in [10], although SRD traffic has an exponential tail, the exponential part of the tail might not become dominant until far beyond the region that is of interest for performance modeling. For example, in [11], realistic examples are given where an MMPP has a "heavy tail" until beyond the point where the probability of blocking is less than 10^{-40}. Beyond that, the exponential component of the tail is dominant but since the model is applied to an ATM traffic example, we are only interested in regions where blocking is no less than 10^{-9}. There has also been some recent work [4], [5] on fitting an MMPP to IP traffic such that the LRD behavior is matched over several orders of magnitude.

Manuscript received November 12, 2001; revised February 23, 2002 and August 27, 2002; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor R. Srikant.

D. P. Heyman is at 101 Lindy Lane, Lincroft, NJ 07738 USA (e-mail: danheyman@yahoo.com).

D. Lucantoni is with DLT Consulting, L.L.C., Ocean, NJ 07712 USA (e-mail: David.Lucantoni@att.net).

Digital Object Identifier 10.1109/TNET.2003.820252

Also, the state-of-the-art in computing performance measures with MMPP models is far beyond that of current LRD modeling. For example, a recent book on the state-of-the-art of LRD modeling was edited by K. Park and W. Willinger. In their overview paper, [6], they summarize the state-of-the-art as illustrated with the following three quotes: "A major weakness of many of the [LRD] queueing based results is that they are asymptotic, in one form or another," "A further drawback of current performance results is that they concentrate on first-order performance measures that relate to (long-term) packet loss rate but less so on second order measures - for example variance of packet loss or delay, generically referred to as jitter - which are of importance in multimedia communication" and finally, "Even less is known about transient performance measures, which are more relevant in practice when convergence to long-term steady-state behavior is too slow to be of much value for engineering purposes." We note that results for the MMPP/G/1 queue are exact, that results exist for moments as well as distributions and that exact results for the transient distributions are also available [8], [12].

This paper is organized as follows. In Section II, we present a method to fit a discrete MMPP (D-MMPP) to an IP data trace. In Section III, we present a method to approximate the effect of limited access line speeds on the traffic. Building upon both of these ideas, we present in Section IV a method for approximating the superposition of D-MMPPs with a D-MMPP having a much smaller state space. This will allow an analytic treatment of the multiplexing of several sources (some of which might represent outputs from previous nodes). Section V contains several numerical examples demonstrating the accuracy of these methods both in terms of approximating the arrival process as well as predicting the performance seen by feeding the process into a queue. Finally, we conclude the paper in Section VI.

II. Fitting an IP Trace

Heffes [13] made the first documented proposal to fit a traffic stream by an MMPP. He used two phases for the Markov chain, gave an estimation procedure for the parameters of the MMPP from count data, and solved several queueing models with exponential service times. Meier-Hellstern [14] provided the first estimation algorithm based on interarrival-time measurements. It has proven difficult to obtain estimators for MMPPs with more than a few states because of the computational burden. Moreover, traffic data frequently consists of counts of events during fixed length intervals, such as packets per second or bytes per 100 ms, so model fitting has to be done with this type of data.

Hidden Markov models take counts as the fundamental random variables, so they are alternatives to MMPP models. A *hidden Markov model* (HMM) is a discrete-time process in which parameters that control the distribution of the number of counts depend on the state of a (hidden) Markov chain [15]. In particular, when the number of counts has a Poisson distribution, we obtain a *Poisson-hidden Markov model* (PHMM). The connection between the MMPP and the PHMM is not straightforward. An MMPP is a point process and the fundamental random variables are the arrival epochs. The phases of an MMPP are governed by a *continuous-time* Markov chain. Given the current phase, the future of the MMPP is conditionally independent of the past (i.e., Markov); this is what makes the MMPP so useful in continuous-time queueing models.

Suppose we define a point process where the phases are governed by a *discrete-time* Markov chain and the rates depend on the phases exactly as in an MMPP. The phase changes can occur at integer multiples of some fundamental time unit; call these times *phase-transition epochs*. This process is not Markovian because the times between phase changes do not have the memoryless property; it is regenerative (at any phase-transition epoch). The essential physics are the same as the MMPP. We propose calling this process a *discrete* MMPP (D-MMPP). A PHMM is a counting process that counts the number of events between adjacent phase-transition epochs of a D-MMPP. It is a special case of a batch Markovian arrival process (BMAP); Lucantoni [8] describes the BMAP and some related processes.

With the counts as data *and the number of states in the Markov chain given*, the log-likelihood function of a PHMM can be written in a computationally useful form. Moreover, it is claimed that extant optimization algorithms can be used to obtain maximum likelihood estimates [15]. The sticking point in this result is that the number of Markov chain states has to be specified in advance, and that may not be easy to do. The claim that maximizing the likelihood function can be done needs to be checked for time series with hundreds of thousands of observations and up to forty states in the Markov chain, which may be required for IP traffic counts. Doing so is beyond the scope of this paper.

Previous approaches to estimating MMPP parameters when there are more than two states use the maximum likelihood estimator (MLE) method [16]. The MLE is also used to estimate parameters of the PHMM [15]. MLEs are computationally expensive because a nonlinear function is minimized; each function evaluation has computational effort that is linear in the length of the trace that is being fitted, and quadratic in the number of states in the Markov chain underlying the MMPP. This may not be tractable for 10 000 observations and 20 states, which is what some of our traffic data yields. (One count every second for three hours produces 10 800 counts). Moreover, the MLE method requires that the number of states of the Markov chain is picked before the calculations are done. Algorithm LAMBDA in Section II-B1 can be used to do this.

The *interrupted Poisson process* (IPP) is the simplest nondegenerate special case of an MMPP. There are two states, and the arrival rate is zero in one of them. Andersson and Ryden [16] developed a maximum likelihood estimation algorithm for MMPPs based on the superposition of IPPs. Their algorithm requires that the arrival times are given in addition to the counts. We want to avoid needing that data. Lee *et al.* [17] describe the properties of a generalized IPP in which the times between phase transitions need not be exponential. They focus on the autocorrelations in the times between arrival epochs. Since only one positive Poisson rate is used, we do not think this model will fit the data we use (see Fig. 1.)

A. D-MMPP Model

The D-MMPP is defined on a discrete Markov chain with $n \times n$ transition probability matrix P. In each discrete step of

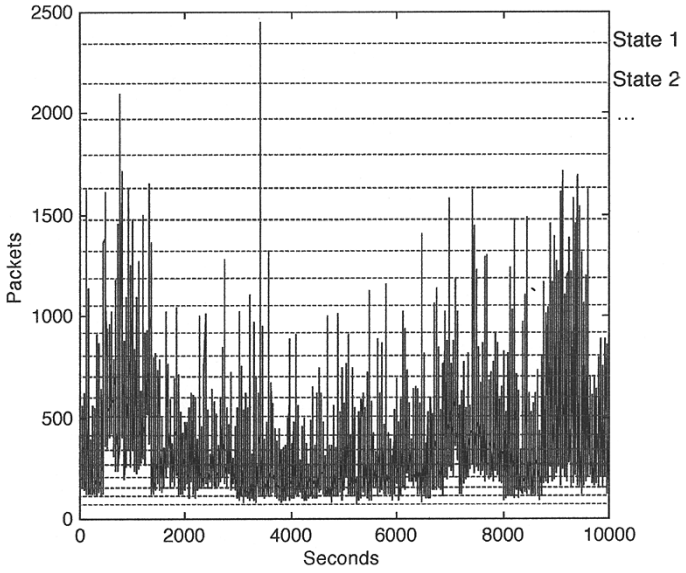

Fig. 1. Sample data trace with the fitted arrival rates.

the Markov chain there will be a Poisson distributed number of arrivals with mean λ_i, where i is the state prior to the transition. Let the arrival rate vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_n)^T$, where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ and n is the number of states of the Markov chain. This definition of a D-MMPP coincides with a discrete BMAP (D-BMAP) with representation $\{D_k\}$ where

$$D_k = \operatorname{diag}\left(e^{-\lambda_1}\lambda_1^k/k!, \ldots, e^{-\lambda_n}\lambda_n^k/k!\right) \cdot P, \tag{1}$$

(see [8]). The irreducible stochastic matrix $D = \sum_{k=0}^{\infty} D_k$ has stationary probability (row) vector $\boldsymbol{\pi}$ and mean arrival rate vector

$$\sum_{k=0}^{\infty} kD_k\boldsymbol{e} = (\lambda_1, \ldots, \lambda_n)^T = \boldsymbol{\lambda} \tag{2}$$

(where $\boldsymbol{e}$ is a column vector of ones) so that the mean arrival rate is $\boldsymbol{\pi\lambda}$.

B. Fitting Data to a D-MMPP

We assume throughout that some data analysis shows that an MMPP or D-MMPP model suitably describes the data. An example of how this may be done is given in Section II-E. As discussed in [18], the reason a two-state MMPP does not describe highly bursty data well is because the union of the ranges of the two Poisson processes is not the range of the data; there is a big gap in the middle. This can be seen from the Normal approximation to the Poisson. For the Normal distribution we have the familiar fact that 95% of the probability is contained within the mean plus or minus 1.96 standard deviations. Since the variance of the Poisson distribution equals the mean (say, λ), and for large λ the Poisson and Normal distribution functions are close, most of the observations of $P(\lambda)$ will be within $\lambda \pm a\sqrt{\lambda}$ with $a = 2$. When the peak-to-mean ratio of the data is large, the two rates in phases one and two, λ_1 and λ_2 say, with $\lambda_1 > \lambda_2$, will be very different. Then $\lambda_2 + 2\sqrt{\lambda_2}$ will be much less than $\lambda_1 - 2\sqrt{\lambda_1}$, and traffic with rates (in the MMPP sense of having different rates persist for random lengths of time) between these values will not be described.

1) Choosing the Rates: Suppose we use rates $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$, where N is not yet specified. We need λ_1 to cover the largest observations, so we choose

$$\lambda_1 + 2\sqrt{\lambda_1} = \text{peak of the data.} \tag{3}$$

Equation (3) leaves some probability beyond the peak of the data, so sample paths of the MMPP can have peaks larger than the peak of the data. This is a quadratic in λ_1 and its solution is

$$\lambda_1 = (\sqrt{1 + \text{peak}} - 1)^2. \tag{4}$$

The lower bound of the data covered by λ_1 is $\lambda_1 - 2\sqrt{\lambda_1}$, which is chosen to be the upper bound of the data covered by λ_2, so we get

$$\lambda_2 + 2\sqrt{\lambda_2} = \lambda_1 - 2\sqrt{\lambda_1}. \tag{5}$$

Equation (5) is a quadratic in λ_2; the solution is

$$\lambda_2 = (\sqrt{\lambda_1} - 2)^2. \tag{6}$$

This procedure can be repeated to obtain λ_3 from λ_2, λ_4 from λ_3, as long as desired. The obvious stopping points are at the minimum of the data or zero.

Four points about this algorithm are worth noting. The first is that the choice of $a = 2$ is essentially arbitrary. A smaller number would yield more rates. The second is that there is some overlap in the values generated by adjacent rates. e.g., the smallest values generated with λ_j are smaller than the largest values generated with λ_{j+1}. Using a smaller number than 2 in the algorithm would yield a larger overlap. The third is that the data requirements are the largest and smallest values of the time series. The fourth is that the differences $\lambda_j - \lambda_{j-1}$ are decreasing; the larger rates are used to cover more potential values.

The algorithm to select the rates is called LAMBDA; a precise statement of the algorithm is given below.

Algorithm LAMBDA

0. Choose width parameter a [default $= 2$] and ending parameter ω [default $= 0$].
1. Choose λ_1 via (4).
   a) Set $N = 1$.
   b) Set $r = \lambda_1 - a\sqrt{\lambda_1}$.
   c) If $r \leq \omega$, Stop; else, go to 2.
2. Set $j = N + 1$.
   a) Set $\lambda_j = (\sqrt{\lambda_{j-1}} - a)^2$.
   b) Set $r = \lambda_j - a\sqrt{\lambda_j}$.
   c) Set $N = j$.
   d) If $r \leq \omega$, Stop; else, repeat step 2.

2) Fitting the Markov Chain: Algorithm LAMBDA yields N rates; we now have to construct a Markov chain transition matrix for the phases. Let $\{x_i, i = 1, 2, \ldots, T\}$ be the observations. We associate x_i with a phase (denoted ϕ_i) as follows:

$$\lambda_j - a\sqrt{\lambda_j} < x_i \leq \lambda_j + a\sqrt{\lambda_j} \Rightarrow \phi_i = j. \tag{7}$$

We interpret $\{\phi_i, i = 1, 2, \ldots, N\}$ as observations on the phase process. Let $P = (p_{ij})$ be the transition matrix for the phase process. The MLE of p_{ij} is

$$p_{ij} = \frac{\text{number of transitions from } i \text{ to } j}{\text{number of transitions out of } i} \tag{8}$$

(see, e.g., [19]).

C. Fitting an N-State MMPP

The (continuous-time) MMPP is better suited as an arrival process in extant algorithms and software for queueing models than is the D-MMPP [8], [9]. In particular, although the D-MMPP/D/1 queue falls into the general "M/G/1" paradigm (see, e.g., [20]), this general structure requires the computation and storage of a large number of matrices which are contained in the transition probability matrix of the underlying Markov chain embedded at transitions. These difficulties have been avoided in the case of a continuous-time Markovian arrival process (MAP); see, e.g., [21].

The phase transitions of an MMPP are governed by a rate matrix (also called an infinitesimal generator) $Q = (q_{ij})$. A way to obtain a Q that "corresponds" to the discrete transition matrix P of a D-MMPP is to choose

$$q_{ij} = p_{ij},\ i \neq j\ q_{ii} = p_{ii} - 1. \tag{9}$$

The properties that makes this choice of Q correspond to P is that both processes have the same mean sojourn time in state i (for every i), probability of any given sequence of states, and steady-state distribution. These assertions follow from basic properties of Markov chains [22]. Putting these into the standard notation for a MAP for use in Section II-D, we have $D_1 = \mathrm{diag}(\boldsymbol{\lambda})$ and $D_0 = Q - D_1$.

Note that one might ask why we first fit a D-MMPP and then convert to a continuous MMPP as opposed to attempting to fit the continuous MMPP directly. The reason is that with high-speed links it is much easier to obtain traffic measurements as counts over discrete intervals than interarrival times of individual packets.

D. Mean Queue Length at Arrivals

Here we will compare the mean delays for the various processes computed using the BMAP/G/1 algorithms with simulated performance results. Let W_V and W_A be the virtual waiting time and the waiting time seen by an arrival, respectively, in the MAP/D/1 queue with mean service time equal to one. Then from [8] and [21] we have

$$E(W_V) = (3\rho - 2\boldsymbol{b}D_1\boldsymbol{e})/(2(1-\rho)) \tag{10}$$
$$E(W_A) = 1 - (\boldsymbol{b}D_1\boldsymbol{e})/\rho + E(W_V)$$

where $\boldsymbol{b} = ((1-\rho)\boldsymbol{g} + \boldsymbol{\pi}D_1)(\boldsymbol{e}\boldsymbol{\pi} + D_0 + D_1)^{-1}$, $\boldsymbol{g}$ is the stationary probability vector of the irreducible stochastic matrix G, which is the unique solution to the matrix functional equation

$$G = e^{D_0 + D_1 G} \tag{11}$$

and $\boldsymbol{\pi}$ is the stationary vector of the infinitesimal generator $D_0 + D_1$. The matrix G is computed from (11) using successive substitution starting with a stochastic matrix (we use $G_0 = \boldsymbol{e}\boldsymbol{\pi}$.) In some cases (presented below), this successive substitution required almost 100 000 iterations, but takes no more than a few seconds on a Macintosh Powerbook G4 to achieve 10 decimal places of accuracy. Using Little's formula [23], the mean queue length at an arbitrary time and at arrivals is given by $E(Q_V) = \rho E(W_V)$ and $E(Q_A) = \rho E(W_A)$, respectively.

E. Example

Fig. 1 shows a sample data trace representing over three hours of traffic on an IP backbone link. A more global view of the data is obtained by looking at a smoothed version of the data. We use the smoother "lowess" as implemented in S-plus [24], with each point of the smoothed version calculated from 0.4% of the data. The plot is shown in Fig. 2. The arrival rate appears to vary among several distinct values, which is consistent with the sample-path behavior of an MMPP.

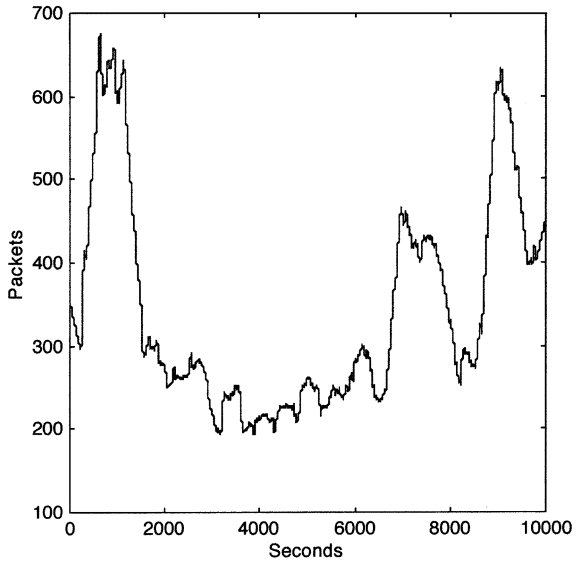

Fig. 2. "Smoothed" version of the data trace.

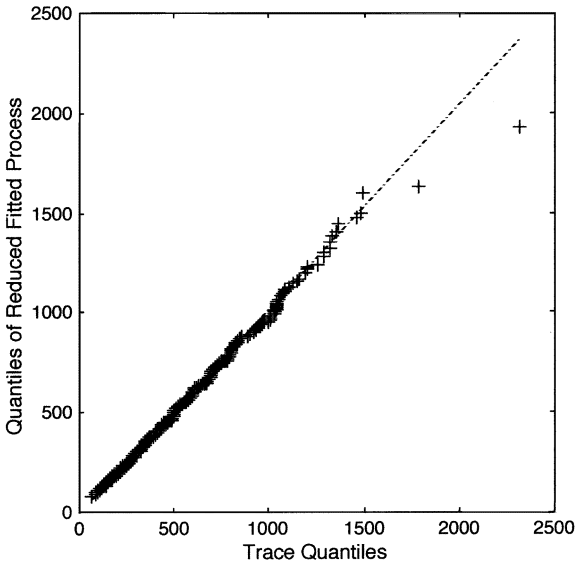

Fig. 3. Q-Q plot of data trace and simulation of the fitted process.

The data measures packets per 500 ms and the mean packet length is 1000 bytes. The mean of the data is 360 packets per 500 ms or 5.76 Mb/s. Algorithm LAMBDA was used to fit this data to a D-MMPP. In this case, 21 states were required and the fitted arrival rates have been superimposed onto the data in Fig. 1. One measure of the goodness of fit of the model is the Quantile-Quantile (Q-Q) plot shown in Fig. 3. Here the quantiles of the data trace and of a simulation of the fitted process are shown. If both sets of data were drawn from the same distribution we would expect the plot to be linear. The fit, as shown, appears to be very good.

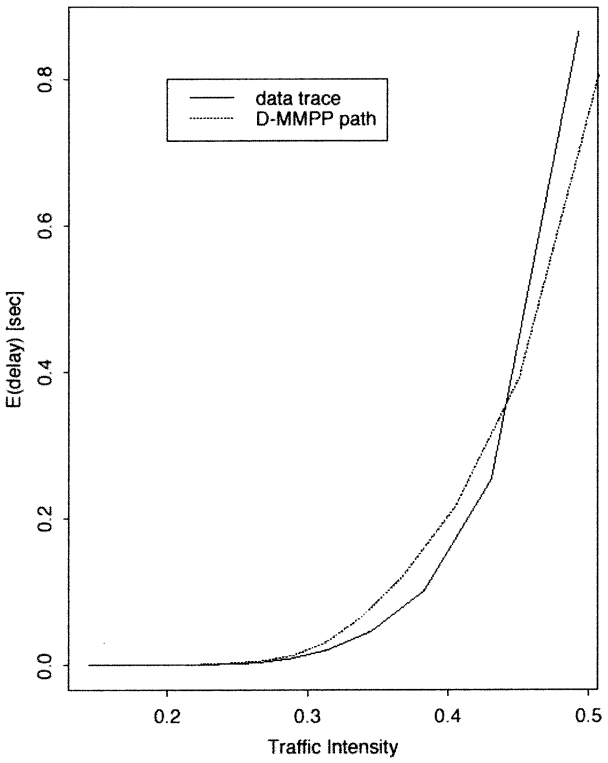

Fig. 4. Comparison of performance of data trace and fitted model.

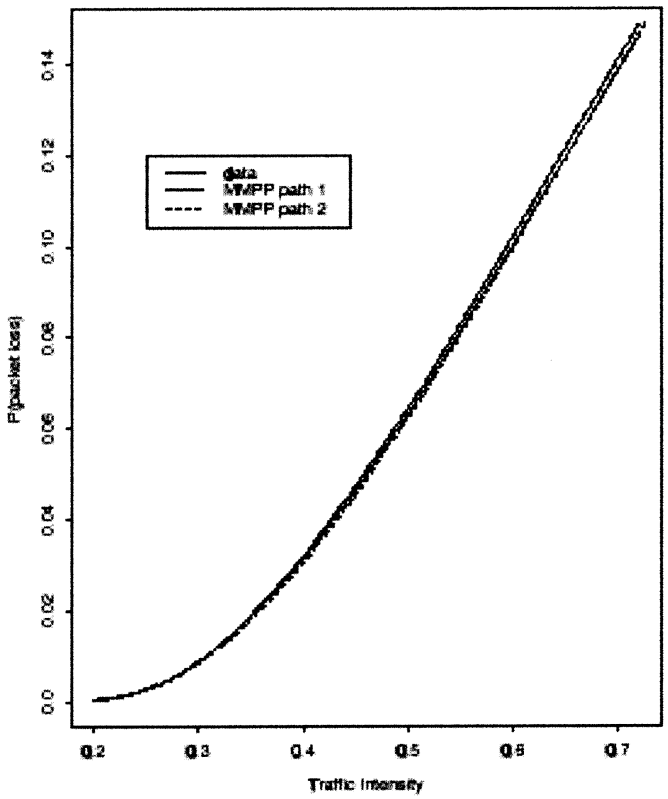

Fig. 5. Comparison of packet loss from data trace and fitted MMPP.

The autocorrelation in the data is not used directly in estimating the parameters of the D-MMPP model; they enter indirectly through the transition matrix. The latter decays to zero much more rapidly than the former. This does not affect mean queue sizes because the data was collected on a link with utilization 0.13 and a simulation at that occupancy yields a mean queue size close to zero for the data and a D-MMPP trace. When the service rate is halved, the mean queue length for the data is 1.63; it is 1.58 for the D-MMPP. Since the data is TCP traffic, it will adapt to congestion and extrapolating to higher utilizations is fraught with danger [25].

To test the ability of the D-MMPP to capture the important features of a packet trace, we estimated the mean packet delay by simulating an infinite buffer queue with constant service times using a packet trace (number of packets per second) as the traffic input. The traffic intensity was varied by changing the service time. We also computed the mean delay from an analytic model where the arrival process was the MMPP fit [see (19) for how an MMPP is obtained from a D-MMPP]. The results are shown in Fig. 4. The mean delays computed from the analytic model and the simulation model are close over the practical region where the mean delay is below 600 ms. The analytic model accurately captures the knee of the curve when the traffic intensity increases from 0.3 to 0.4. At higher traffic intensities (above 0.5), the mean delay from the simulation is significantly larger than the mean delay from the analytic model. It does not appear that the correlations in the data have a significant impact on the ability of the D-MMPP to provide accurate estimates of queueing performance when the delays are in the practical range. The fact that accuracy declines at higher traffic intensities is consistent with caveats about extrapolating a TCP trace to higher traffic intensity issued by Arvidsson and Karlsson [25].

A second test of the goodness of fit is a comparison of packet loss probabilities when there is a finite buffer. The maximum delay in a router is controlled by limiting the size of the buffer. Our data averages 360 packets per half-second, so when $\rho = 0.2$ (processing rate is 1800 pps) it takes 2.8 ms to empty a buffer of size 10, and when $\rho = 0.72$ (processing rate is 500 pps) it takes 10 ms to empty that buffer. These are typical design values. Fig. 5 shows the packet loss probabilities for simulations driven by the data trace and by two realizations of the MMPP fitted to the trace. The curves are very close, so the MMPP captures the relevant features of the trace. Similar closely matched curves were obtained for a buffer of size 100. In those simulations, the distributions of queue length were very close; they are not shown because virtually all of the probability is concentrated at two values: 0 and 100.

III. Effect of Limiting Access Line Speeds

It is well known that the Poisson process is not bursty enough to adequately describe the type of data seen on IP links. In attempting to demonstrate how poorly the Poisson distribution fits the data, we displayed data generated by the Poisson distribution with the same mean as the data side-by-side to a portion of the original data, as seen in Fig. 6.

It is immediately apparent that with respect to the high rates that we observe in IP backbone traffic, the Poisson distribution is relatively constant. Therefore, assuming that the original traffic is adequately modeled by a D-MMPP, it is simple to represent the same process transmitted over an access line with a limit

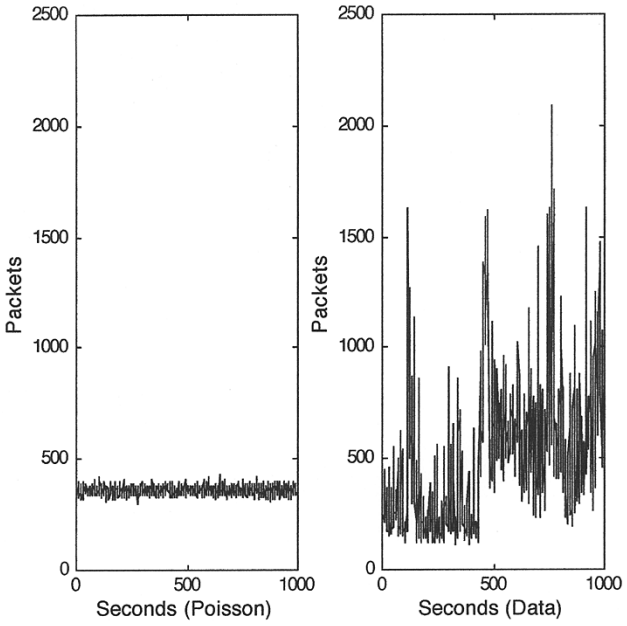

Fig. 6. Poisson process with the same mean rate as the data trace.

on the transmission rate as another D-MMPP where all states of the original process with corresponding rates higher than the peak rate will be collapsed into one state with the peak rate as its arrival rate. This result is stated in Theorem 1.

Consider a D-MMPP with transition probability matrix P and arrival rate vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_n)^T$, where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. If the D-MMPP is being used to approximate a bursty traffic stream entering an output port, multiplexer, etc., as the traffic intensity increases the peak rate of the stream will be limited by the rate of the link. We now provide a method for modifying the parameters of the D-MMPP so that the new model captures the effect of limiting the peak. Let $\boldsymbol{\pi}$ be the stationary vector of the transition matrix P. That is

$$\boldsymbol{\pi} = \boldsymbol{\pi} P, \quad \boldsymbol{\pi} \boldsymbol{e} = 1 \tag{12}$$

where $\boldsymbol{e} = (1, 1, \ldots, 1)^T$. Then the mean arrival rate of the D-MMPP is $\lambda^* = \boldsymbol{\pi}\boldsymbol{\lambda}$. Let the peak rate of the input line be $\bar{\lambda}$. Suppose $\lambda_1 > \bar{\lambda}$, and (for the moment), that all the other λ's are less than $\bar{\lambda}$. Let T_1 denote a generic sojourn time in state 1 of the Markov chain. When the Markov chain is in state 1, packets arrive faster than the input line can process them. The expected number of arrivals during a sojourn in state 1 is

$$\lambda_1 E(T_1) = \lambda_1/(1 - P_{11}). \tag{13}$$

Assume the excess packets are stored in a buffer and that the buffer is large enough that packet losses are negligible. (A way to incorporate packet loss would be to set the rate of the output process to the carried load on the link. The carried load can be computed by standard queueing models of the given link and buffer size.) The actual rate that the packets flow on the input line is $\bar{\lambda}$, so to conserve the expected number of packets during a sojourn time in a state, we change the D-MMPP model by replacing λ_1 by $\bar{\lambda}$ and P with P' which is chosen to maintain the mean arrival rate. We also preserve the behavior during sojourns in states corresponding to rates lower than the peak. Since $\lambda_1 > \bar{\lambda}$ and the other λ's are held fixed, $P'_{11} > P_{11}$; i.e., the mean sojourn time in state 1 is increased. Now we show how to derive the appropriate parameters of a D-MMPP to approximate this behavior.

Let m be the index for which $\lambda_m \geq \bar{\lambda}$ and $\bar{\lambda} > \lambda_{m+1}$. Let $k = n - m$ and partition P and $\boldsymbol{\lambda}$ as

$$P = \begin{bmatrix} S & U \\ V & T \end{bmatrix}, \quad \boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_S \\ \boldsymbol{\lambda}_T \end{bmatrix} \tag{14}$$

where $S = m \times m$, $T = k \times k$, and $\boldsymbol{\lambda}_S$ and $\boldsymbol{\lambda}_T$ are $m \times 1$ and $k \times 1$, respectively. We would like to collapse the states corresponding to rates greater than or equal to the peak rate into one state with the peak rate as its corresponding rate and leave the behavior of the other states unchanged. The sojourn time in the peak-rate state should be extended so that the overall average rate is unchanged. We also partition the stationary vector as $\boldsymbol{\pi} = (\boldsymbol{\pi}'_S, \boldsymbol{\pi}'_T)$ and normalize the first component to be $\boldsymbol{\pi}_S = \boldsymbol{\pi}'_S/\boldsymbol{\pi}'_S\boldsymbol{e}$.

Theorem 1: The D-MMPP with the desired properties has a transition probability matrix given by

$$P' = \begin{bmatrix} 1-p & p\boldsymbol{u} \\ \boldsymbol{v} & T \end{bmatrix} \tag{15}$$

with mean arrival rate vector $\boldsymbol{\lambda}' = (\bar{\lambda}, \boldsymbol{\lambda}_T^T)^T$, where

$$\begin{aligned} &\boldsymbol{u} = \boldsymbol{\pi}_S(I-S)^{-1}U/\boldsymbol{\pi}_S(I-S)^{-1}U\boldsymbol{e} \\ &\boldsymbol{v} = V\boldsymbol{e}, \quad p = (\lambda^* - \bar{\lambda})/(\beta\lambda^* - \gamma) \\ &\beta = -\boldsymbol{u}(I-T)^{-1}\boldsymbol{e} \quad \text{and} \quad \gamma = -\boldsymbol{u}(I-T)^{-1}\boldsymbol{\lambda}_T. \end{aligned} \tag{16}$$

Proof: Clearly, the D-MMPP with representation given in (15) has peak rate, $\bar{\lambda}$, and the behavior when the rate is below the peak is the same as that in the original D-MMPP. The vector $\boldsymbol{v}$ is chosen so that the rate into the peak state from less-than-peak-rate state i is the same as the rate into any of the states greater than or equal to the peak state in the original process. The probability of entering the less-than-peak-rate states for the first time in state j, given that the process starts in the greater-than-peak-rate state i is given by the (i, j) element of the matrix $(I - S)^{-1}U$ (see [26] or [27, Lemma 1.C]).

By direct calculation, we can show that the stationary probability vector of the matrix P' is given by

$$\boldsymbol{\alpha} = (1 + p\boldsymbol{u}(I-T)^{-1}\boldsymbol{e})^{-1}(1, p\boldsymbol{u}(I-T)^{-1}) \tag{17}$$

and that the average arrival rate is $\boldsymbol{\alpha}\boldsymbol{\lambda}'$. Setting this equal to λ^* and solving for p gives the final expression in the theorem.

Note that the essence of this result is a technique for collapsing a group of states into a single state (i.e., all states with rates greater than the peak rate are collapsed into a single state corresponding to the peak rate.) This idea will be utilized in Section IV on approximating a multiplexed process.

A. Example

The first graph in Fig. 7 shows a simulation of the original data using the fitted D-MMPP computed earlier. Recall that this process had a mean of 5.76 Mb/s. If this traffic is transmitted over a 45-Mb/s access line, then there will be very little effect on the traffic as seen by the dotted line in the first graph. The second and third graphs in the figure show the traffic scaled by 4 and 6, respectively, with link capacity held at 45 Mb/s. That is, the second graph represents traffic with a mean rate of 23 Mb/s, and the third has a rate of 34.6 Mb/s.

We clearly see that by limiting the peak rate we achieve a certain level of smoothing. Note that in order to achieve the same

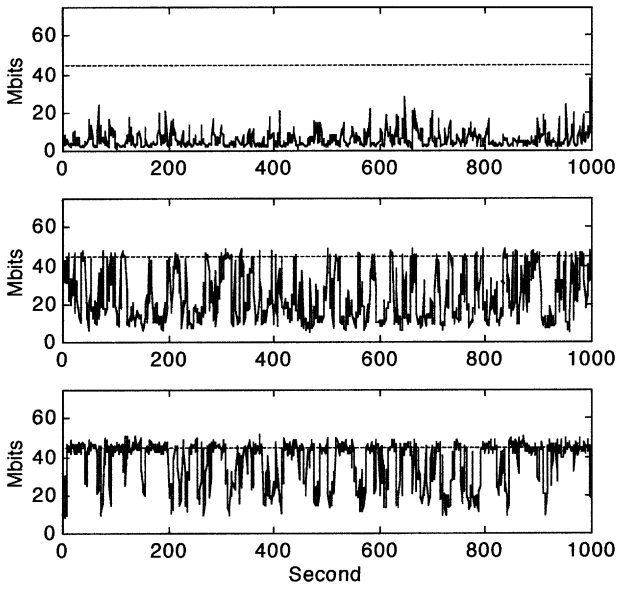

Fig. 7. Simulated data traces with mean rates 5.76, 23, and 34.6 Mb/s with a 45-Mb/s limit.

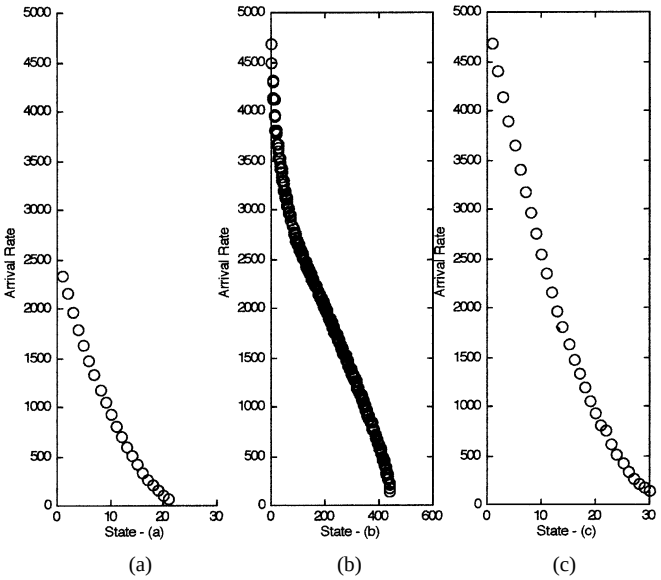

Fig. 8. Arrival rates of (a) the single process, (b) the multiplexed process, and (c) the reduced process.

average rate of the original process, as the peak becomes more limiting, the D-MMPP must remain at the peak rate for longer times.

We also note that we are modeling the output process of a node and not attempting here to model the delay through that node. Clearly, an end-to-end model would need to incorporate the delay (and blocking) seen by specific packets traversing the network. Such an analysis could be done by incorporating the traffic models discussed in this paper into performance models. That analysis is beyond the present scope.

IV. Modeling Multiplexed Traffic

If two traffic streams have been adequately modeled by D-MMPPs with representations $(P_1, \boldsymbol{\lambda}_1)$ and $(P_2, \boldsymbol{\lambda}_2)$, respectively, then the process obtained by multiplexing these streams is given by the superposition of these two processes. It is well known [8] that the superposition is again a D-MMPP with arrival rate vector $\boldsymbol{\lambda}$ given by

$$\boldsymbol{\lambda} = \mathrm{diag}(\Delta(\boldsymbol{\lambda}_1) \oplus \Delta(\boldsymbol{\lambda}_2)) \tag{18}$$

and transition probability matrix P given by

$$P = P_1 \otimes P_2 \tag{19}$$

where $\otimes$ represents the Kronecker product, $\oplus$ represents the Kronecker sum [28], Δ forms a diagonal matrix with its argument on the diagonal and diag extracts the diagonal of a matrix in column vector form. If $\boldsymbol{\lambda}_1$ is of order m_1 and $\boldsymbol{\lambda}_2$ is of order m_2, then the order of the superposition process is $m_1 \times m_2$. This explosion in the size of the state space has been the main hindrance to numerical solutions to the queueing model with multiplexed MMPP arrivals.

We make several observations. First, the reason the state space is so large is that every possible combination of states in the original processes needs to be treated. If the original processes were fit to data using Algorithm LAMBDA, then by definition, the rates are spread out over the range of the data. To simplify notation, let us assume that we are multiplexing two identical streams with parameters $(P, \boldsymbol{\lambda})$ where $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_n)^T$ ordered from largest to smallest. Then the largest rate in the superposed process is $2\lambda_1$ and the smallest rate is $2\lambda_n$ and all of the other $n^2 - 2$ rates will be distributed between these two extremes.

In Fig. 8, we show the original arrival rates of the single process along with the arrival rates associated with the superposition process. It is clear that the multiplexed process has many states where the rates are very close together. Applying the reasoning of Section II, we construct another D-MMPP with a very similar behavior with a much smaller set of states whose associated rates are spread out to cover the original range of states. In particular, starting with the peak rate, we apply Algorithm LAMBDA to compute a sequence of rates with the appropriate overlap governed by the spacing parameter a. In Fig. 8(c), we display the rates computed using $a = 2$. Note that only 30 states are required to approximate the multiplexed process with a total of $21 \times 21 = 441$ states. The effectiveness of the approximation will be discussed in Section V.

A. Reduction Algorithm

We now present an algorithm for approximating the superposition of two D-MMPPs with a D-MMPP with a much smaller state space. Given two D-MMPPs with parameters $(P_1, \boldsymbol{\lambda}_1)$ with n_1 states and $(P_2, \boldsymbol{\lambda}_2)$ with n_2 states, respectively, we first compute the actual parameters $(P, \boldsymbol{\lambda})$ of the superposed process from (18) and (19). Let $n = n_1 n_2$ be the number of states. We then reorder the states so that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. Let the parameters of the approximating process be denoted by $(P', \boldsymbol{\lambda}')$. The approximating process retains the largest arrival rate, i.e., $\lambda'_1 = \lambda_1$. We then apply one step of Algorithm LAMBDA to compute the next lower rate, λ'_2, using the spacing parameter a. We then reorder the states so that λ'_1 is placed at the bottom of the vector $\boldsymbol{\lambda}'$ and then apply the peak-limiting algorithm of Section III with "peak" rate λ'_2. Specifically, with the above notation we have the following algorithm.

Algorithm REDUCTION

0. Compute the stationary vectors $\{\boldsymbol{\pi}_i, i = 1, 2\}$, of the original D-MMPPs, i.e.,

$$\boldsymbol{\pi}_i P_i = \boldsymbol{\pi}_i, \qquad \boldsymbol{\pi}_i \boldsymbol{e} = 1, \qquad i = 1, 2. \tag{20}$$

1. Compute the stationary probability vector of the superposed process $\boldsymbol{\pi}' = \boldsymbol{\pi}_1 \otimes \boldsymbol{\pi}_2$ and the average rate of the superposed process $\lambda^{*\prime} = \boldsymbol{\pi}_1 \boldsymbol{\lambda}_1 + \boldsymbol{\pi}_2 \boldsymbol{\lambda}_2$.

2. Choose width parameter $a[\text{default} = 2]$.

3. Set $\lambda'_1 = \lambda_1, \ P' = P$.

4. Reorder the states so that $\boldsymbol{\lambda}' = (\lambda_2, \lambda_3, \ldots, \lambda_n, \lambda'_1)^T$ and adjust the transition matrix P' to reflect the same state reordering. Set $i = 0$.

5. Continue

a) Set $i = i + 1$.

b) Set $r = \lambda_i - a\sqrt{\lambda_i}$.

c) Set $\lambda'_{i+1} = (\sqrt{r + a/2} - a/2)^2$.

d) Set $\text{peak} = \lambda'_{i+1}$ and compute a new version of $\boldsymbol{\lambda}'$ and P' according to Theorem 1. Note that, for some values, there is no feasible value p (from Theorem 1) corresponding to the rate peak that will result in the same overall average rate. In this case, we set $p = 1$, corresponding to remaining in the corresponding state for just one step, and then compute the corresponding rate needed to achieve the original average rate. That is, from the second equation in (16), we have

$$\lambda'_{i+1} = \lambda^* + \beta\lambda^* - \gamma. \tag{21}$$

e) If $\lambda'_{i+1} \leq \lambda_{1_{n_1}} + \lambda_{2_{n_2}}$, stop; else, go to step 6.

6. Let j be the index for which $\lambda_{j-1} > \lambda'_{i+1} > \lambda_j$. Reorder the states so that $\boldsymbol{\lambda}' = (\lambda_j, \lambda_{j+1}, \ldots, \lambda_n, \lambda'_1, \ldots, \lambda'_{i+1})$ and adjust the transition matrix P' to reflect the same state reordering. Go to step 5.

V. Numerical Examples

In this section, we present some examples of the previous results. In particular, we start with the D-MMPP approximation of the data trace discussed in Section II-E. We show that we can approximate a superposition of four of these processes (which would require over 194 000 states to be modeled exactly) with a process with just 41 states.

A. Multiplexed Process

In order to demonstrate the effectiveness of the multiplexing algorithm, we first generate four independent simulated traces from the fitted D-MMPP process. These are labeled Trace A–D, respectively, in Fig. 9. The initial state of each trace is chosen according to the stationary probability vector of the D-MMPP.

We can then superpose these simulated traces to achieve a valid trace from the superposition of the D-MMPPs. This is much more efficient than forming the D-MMPP representation of the superposition and simulating from that. In fact, due to the tremendous increase in the state space of the superposition, that approach would not be feasible. In Fig. 10, we show the multiplexed streams Trace A + Trace B (labeled Trace AB) and Trace C + Trace D (labeled Trace CD).

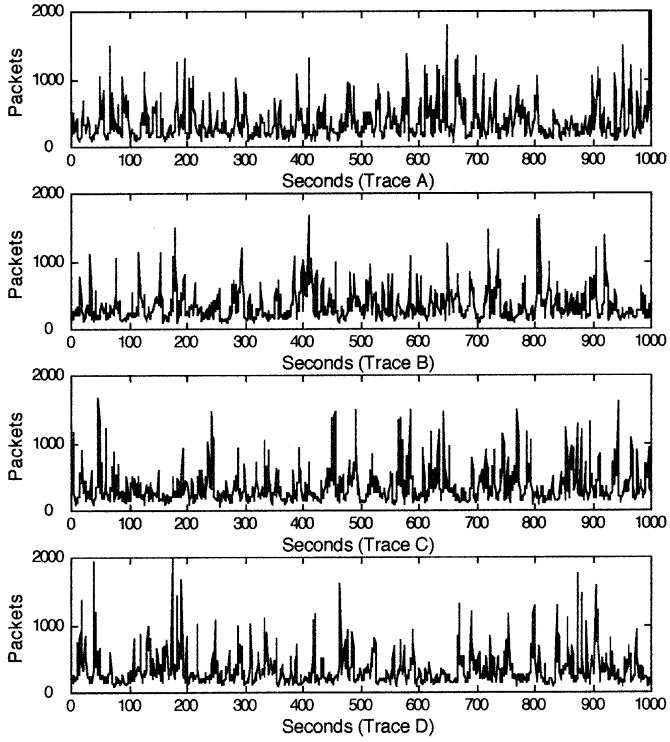

Fig. 9. Four simulated traces from D-MMPP-1.

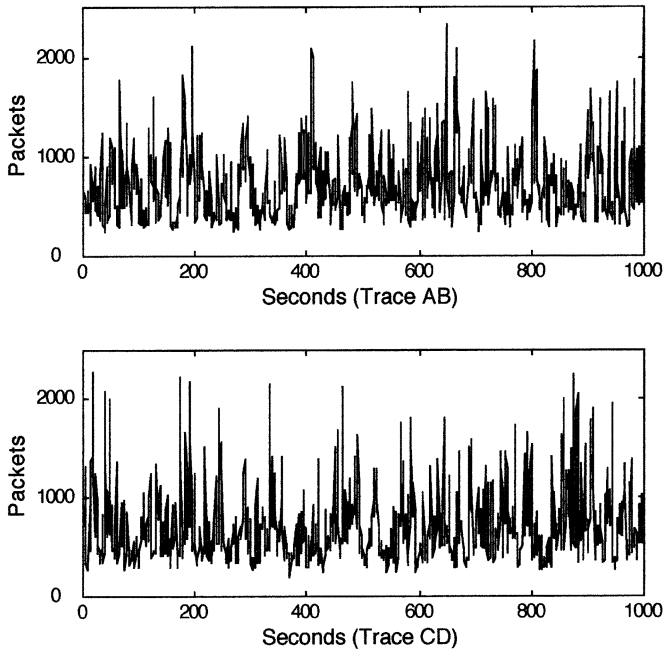

Fig. 10. Multiplexed traces A + B and C + D.

The representation of the D-MMPP corresponding to the superposition of two identical processes represented by the original D-MMPP is obtained from (18) and (19) and in this case has $21 \times 21 = 441$ states. From this representation we apply the REDUCTION algorithm to obtain an approximate representation (D-MMPP-2) of the superposed process. In this example, the reduced process has 31 states. Fig. 11 shows the Q-Q plot of Trace AB with a simulated trace from the reduced D-MMPP-2. Once again, we see a very good fit.

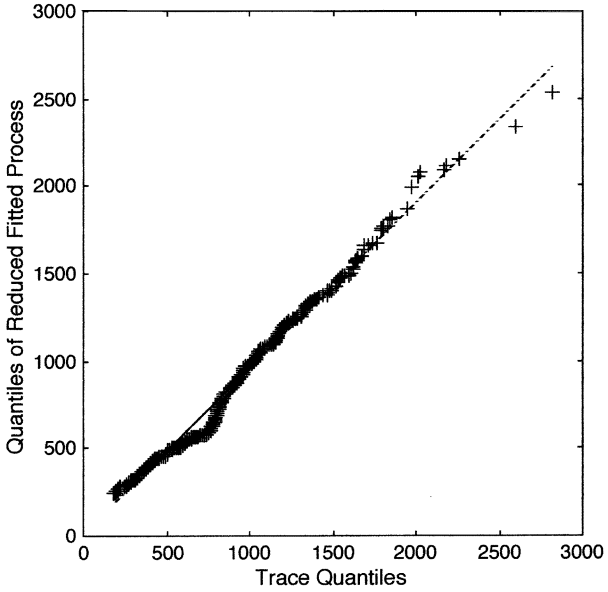

Fig. 11. Q-Q plot of the sum of Trace A + Trace B and the reduced fitted process.

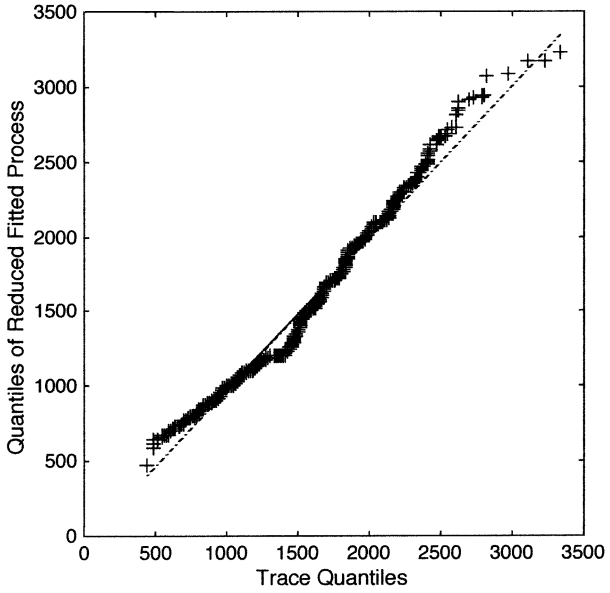

Fig. 13. Q-Q plot of the sum of four traces and a simulation of the reduced fitted process.

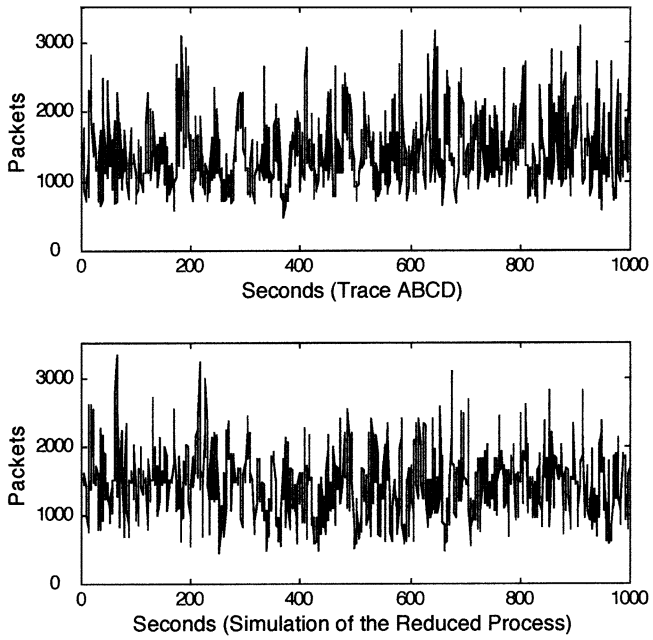

Fig. 12. Sum of four traces and a sample trace of the reduced fitted process.

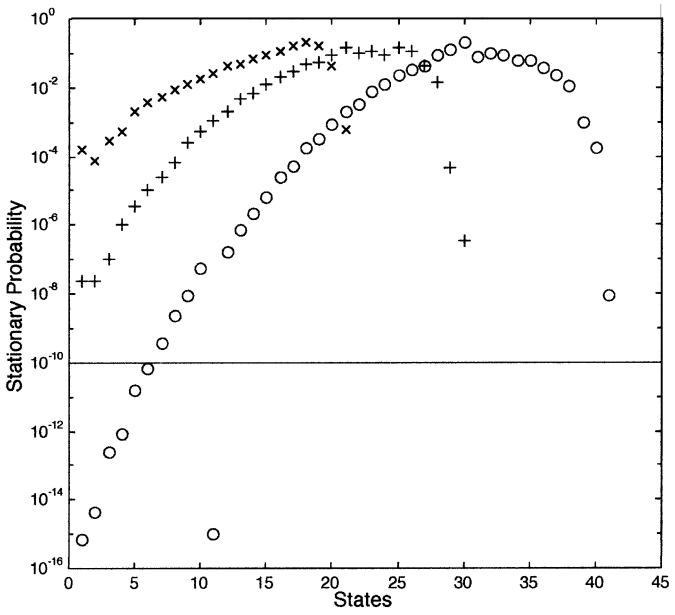

Fig. 14. Stationary probabilities for 1-, 2-, and 4-stream models.

Finally, we present the multiplexed process obtained by summing all four traces (A–D) in Fig. 9. Note that the number of states in the exact representation of the D-MMPP representing this process would require $21^4 = 194\,481$ states. This is far beyond our capabilities of obtaining exact performance results from existing performance algorithms.

Our approximation starts with the 31-state reduced D-MMPP-2. The superposition of this approximating process would contain $312 = 961$ states. We then apply the REDUCTION algorithm to this superposition and obtain an approximate representation (D-MMPP-4) with 41 states. A simulation of the approximate process is shown in Fig. 12 along with the superposition of the four original traces.

The goodness-of-fit for the one-dimensional distributions of the superposition is shown by the Q-Q plot in Fig. 13. To our eyes, the fit appears to be "good enough" to use the approximate superpositions instead of the exact representation in queueing models.

B. Further Reduction in State Space

In Fig. 14, we show the stationary probabilities of the states in the 1-stream, 2-stream and 4-stream models. If we eliminate the states with a stationary probability smaller than 10^{-10} then we can reduce the number of states from 41 to 34 in the 4-stream model. Note that this approximation would normally eliminate the state corresponding to the sum of the peaks. This is not a bad thing. Since the probability of both processes being in the peak

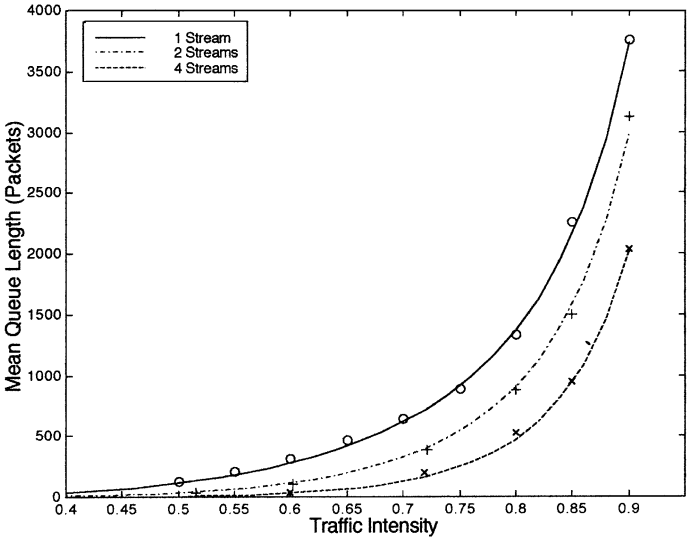

Fig. 15. Analytic and simulation results for queue length at arrivals.

state at the same time is negligible, we need not track that state for the approximation. In fact, this also explains the smoothing that is accomplished by superposing the processes.

C. Performance Comparisons

In Fig. 15, we plot the analytic and simulation results for the mean queue length at arrivals for one, two, and four streams. The analytic results are well within the 95% confidence intervals of the simulation results.

VI. Conclusion

We have shown how to estimate the rates of a D-MMPP with Algorithm LAMBDA, and how to estimate the transition matrix with (7). The modifications in the parameters of a D-MMPP that can reflect the effects of a rate-limiting server are given by Theorem 1. The algorithm REDUCTION provides an approximate D-MMPP for the superposition of two independent D-MMPPs. An empirical demonstration that our procedures provide a good match to simulations was given. By combining the approximation for the rate-limited arrival process and the approximation of the multiplexed process, one can explicitly model traffic coming to a backbone router from inhomogeneous sources which first go through access routers. The state-space explosion that accompanies exact modeling is avoided at the expense of slight approximation inaccuracy. Simulation is unlikely to be a viable alternative to an approximate analytic model. Simulating the superposition of MMPPs is very time consuming. If it is done by simulating the phase changes from the exact transition matrix, a large number of arrivals have to be simulated to ensure that the phase process has enough transitions to give a representative path. If it is done by simulating the constituent MMPPs and combining them, several paths have to be generated and combined. The simulations reported in Section V-C for the superposition of four D-MMPPs were done that way, and took about two hours on a SUN Ultra-Enterprise 3000.

The approximation of the D-MMPP given by algorithm LAMBDA will not provide an appropriate set of conditional arrival rates when the peak rate is too small. For count data, this means that the interval during which the counts are made cannot be so short that the variability of the data is insignificant. To see this, consider the extreme case when the length of an interval is the time to process one bit. Then each interval has either zero or one bit, which is not an MMPP because when one event occurs for certain, the number of events does not have a Poisson distribution. As the length of the measurement interval is increased from this tiny value, the number of events in an interval will start to exhibit nondegenerate stochastic fluctuations, and an MMPP may provide a suitable model.

The same analysis carries over to the approximation of the superposition of D-MMPPs; our approximation will not work when the peak rate is too small. However, under this condition the D-MMPP will not have many states, and the superposition of two of them can be done exactly without extraordinary computation. For example, we did an example where the original D-MMPP had five states. The approximation of the superposition attempted to use seven states, and this was not a good approximation of the true process which has 25 states (and was not difficult to compute with). One can attempt to approximate the superposition of four of these D-MMPPs by the approximate method, which is worth being able to do because the exact representation has 625 states. Investigating the range of applicability of our approximations will be the subject of future work.

Acknowledgment

The authors would like to thank Y. Levy, K. Meier-Hellstern, P. Wirth, and the referees for their comments on this paper.

References

[1] H. Heffes and D. M. Lucantoni, "A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance," *IEEE J. Select. Areas Commun.*, vol. SAC-4, pp. 856–868, Sept. 1986.

[2] C. Blondia and T. Theimer, "A discrete-time model for ATM traffic," RACE, Doc. PRLB-123-0018-CD-CC/UST-123-0022-CD-CC, 1989.

[3] M. F. Neuts, "Modeling data traffic streams," in *Teletraffic and Datatraffic in a Period of Change*, A. J. a. V. B. Iversen, Ed. Amsterdam, The Netherlands: North-Holland, 1991.

[4] A. Andersen and B. Nielsen, "A Markovian approach for modeling packet traffic with long range dependence," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 719–732, June 1998.

[5] P. Salvador, R. Valadas, and A. Pacheco, "Multiscale fitting procedure using Markov-modulated Poisson processes," *Telecommun. Syst. J.*, vol. 23, pp. 123–148, June 2003.

[6] K. Park and W. Willinger, "Self-similar network traffic: An overview," in *Self-Similar Network Traffic and Performance Evaluation*, K. Park and W. Willinger, Eds. New York: Wiley, 2000.

[7] J. Cao, W. S. Cleveland, D. Lin, and D. X. Sun, "On the nonstationarity of Internet traffic," in *Proc. ACM SIGMETRICS*, 2001, pp. 102–112.

[8] D. M. Lucantoni, "The BMAP/G/1 Queue: A Tutorial," in *Models and Techniques for Performance Evaluation of Computer and Communications Systems*, L. Donatiello and R. Nelson, Eds. New York: Springer-Verlag, 1993, pp. 330–58.

[9] G. Latouche and V. Ramaswami, "Introduction to matrix analytic methods," in *Proc. SIAM_ASA Stochastic Modeling Conf.*, 1999.

[10] D. Lucantoni, "After long range dependency (LRD) discoveries, what are the lessons learned so far to provide QoS for Internet advanced applications," presented at the Int. Teletraffic Congr. Panel Discussion, Salvador da Bahia, Brazil, 2001.

[11] G. L. Choudhury, D. M. Lucantoni, and W. Whitt, "Squeezing the most out of ATM," *IEEE Trans. Commun.*, vol. 44, pp. 203–17, Feb. 1996.

[12] D. M. Lucantoni, G. L. Choudhury, and W. Whitt, "The transient BMAP/G/1 queue," *Stochastic Models*, vol. 10, no. 1, pp. 145–82, 1994.

[13] H. Heffes, "A class of data traffic processes—Covariance function characterization and relating queueing results," *Bell Syst. Tech. J.*, vol. 59, pp. 437–488, 1980.
[14] K. Meier-Hellstern, "A fitting algorithm for Markov-modulated Poisson processes having two arrival rates," *Eur. J. Oper. Res.*, vol. 29, pp. 370–377, 1987.
[15] I. L. MacDonald and W. Zucchini, *Hidden Markov and Other Models for Discrete-Valued Time Series*. London, U.K.: Chapman-Hall, 1997.
[16] S. Andersson and T. Ryden, "Maximum likelihood estimation of a structured MMPP with applications to traffic modeling," presented at the 13th ITC Specialist Seminar, Monterey, CA, 2000.
[17] Y. D. Lee, A. van de Liefvoort, and V. L. Wallace, "Modeling Correlated Traffic with a Generalized IPP," *Perform. Eval.*, vol. 40, no. 1–3, pp. 99–114, 2000.
[18] D. Heyman, "Estimation of MMPP models of IP traffic," presented at the 11th INFORMS Applied Probability Soc. Conf., New York, NY, 2001.
[19] P. Billingsley, *Statistical Inference for Markov Processes*. Chicago, IL: Univ. of Chicago Press, 1961.
[20] M. F. Neuts, *Structured Stochastic Matrices of M/G/1 Type and Their Applications*. New York: Marcel Dekker, 1989.
[21] D. M. Lucantoni, "New results for the single server queue with a batch Markovian arrival process," *Stochastic Models*, vol. 7, pp. 1–46, 1991.
[22] D. P. Heyman and M. J. Sobel, *Stochastic Models of Operations Research, Vol. I*. New York: McGraw-Hill, 1982.
[23] J. D. C. Little, "A proof for the formula $L = \lambda W$," *Oper. Res.*, vol. 14, pp. 723–27, 1961.
[24] W. N. Venables and B. D. Ripley, *Modern Applied Statistics with S-Plus*. New York: Springer-Verlag, 1994.
[25] A. Arvidsson and P. Karlsson, "On traffic models for TCP/IP," in *Proc. ITC 16*, P. Key and D. Smith, Eds., 1999, pp. 455–466.
[26] J. Kemeny and J. L. Snell, *Finite Markov Chains*. Princeton, NJ: Van Nostrand, 1960.
[27] D. P. Gaver, P. A. Jacobs, and G. Latouche, "Finite birth-and-death models in randomly changing environments," *Adv. Appl. Probabil.*, vol. 16, pp. 715–31, 1984.
[28] A. Graham, *Kronecker Products and Matrix Calculus with Applications*. London, U.K.: Ellis Horwood, 1981.

Daniel P. Heyman received the B.S. degree in electrical and industrial engineering from Rensselaer Polytechnic Institute, Troy, NY, in 1960 and the M.S. and Ph.D. degrees in operations research from Syracuse University, Syracuse, NY, in 1962 and the University of California at Berkeley in 1966, respectively.

He joined Bell Laboratories in 1966, moved to Bellcore when it was formed in 1984, and joined AT&T Laboratories in 1997. He was a Principal Technical Staff Member in the Network Design and Performance Analysis Department, AT&T Laboratories, Middletown, NJ. He retired in April, 2003. He has published several dozen papers on various topics in applied probability, and is a coauthor of the two-volume book *Stochastic Models in Operations Research* (New York: McGraw-Hill, 1982 and 1984), which received an Honorable Mention in the 1982 ORSA Frederick W. Lanchester Prize for the best publication in operations research.

Dr. Heyman was a corecipient of the 1995 ACM SIGMETRICS/Performance Conference Outstanding Paper Award.

David Lucantoni (M'94–SM'99) received the B.S. degree in mathematics from Towson University, Baltimore, MD, in 1976, and the M.S. degree in statistics and the Ph.D. degree in operations research from the University of Delaware, Newark, in 1978 and 1981, respectively.

He joined Bell Telephone Laboratories in 1981 and for the next 13 years worked on the performance analysis of various telecommunication systems. He has published over 50 professional papers ranging from multiplexer design to broadband congestion control algorithms to state-of-the-art solution techniques to complex stochastic models. He then worked at Motorola's Satellite Communications Division as a performance analyst for the IRIDIUM™ Low Earth Orbit satellite system. Since 1995, he has been an independent consultant in the telecommunications industry and has recently been involved as an expert witness for several litigations. He is currently the Principal Consultant for DLT Consulting, L.L.C., Ocean, NJ (http://www.dltconsulting.com).

Dr. Lucantoni was the corecipient of the 1986 IEEE Stephen O. Rice Prize Paper Award in the field of communications theory. He received an Honorable Mention in the 1998 INFORMS Frederick W. Lanchester Prize for the best publications in the areas of operations research and management science. He also received the 2000 Lucent Patent Recognition Award for patents that were "of significant importance to Lucent Technologies' commercial success" for the patent that proposed the use of cell monitoring and tagging and selective cell discard that is now part of the ATM standards (e.g., CLP bit, UPC, selective discard).