Performance Analysis of an Asynchronous Multi-rate Crossbar with Bursty Traffic

Paul Stirpe*and Eugene Pinsky†
Computer Science Department
Boston University
Boston, MA 02215
email: stirpe@cs.bu.edu
email: pinsky@cs.bu.edu

Abstract

One of the most promising approaches to building high speed networks and distributed multiprocessors is the use of optical interconnections. The basic component of such a system is a switch (interconnection network) that has a capacity of interconnecting a large number of inputs to outputs. In this paper we present an analysis of an $N_1 \times N_2$ asynchronous crossbar switch model for all-optical circuit-switching networks that incorporates multi-rate arrival traffic with varied arrival distributions. We compare the model behavior using traffic loads derived from the Binomial, Pascal, and Poisson statistical distributions. We give efficient algorithms to compute the performance measures. We analyze the effect of load changes from particular traffic distribution streams on system performance and give a simple "economic" interpretation.

1 Introduction

One of the most promising approaches to building high speed networks and distributed multiprocessors is the use of optical interconnections. The basic component of such a system is a switch (interconnection network) that has a capacity of interconnecting a large number of inputs to outputs.

In this paper we present an analysis of an $N_1 \times N_2$ asynchronous crossbar switch model for all-optical circuit-switching networks.

The performance analysis of crossbar switches has been motivated by telephone switching systems [2] and by the development of multiprocessor computer systems [3, 5, 26]. Recent developments towards wideband switches for future visual and data communications [18] have spurred increased interest in the design and analysis of interconnection networks operating in asynchronous mode. A practical implementation of an asynchronous crossbar is described in [6]. A recent survey of performance results including congestion control schemes is given in [25]; for other interconnection networks see [1, 4, 12, 14, 21, 22, 23, 31, 32, 34].

The electrical crossbar switch is a basic building block in the design of many switching systems. Such a switch is internally non-blocking and all electrical paths experience constant end-to-end delay. However, the circuit complexity of an $N \times N$ crossbar is $O(N^2)$. As a result, researchers proposed Multistage Interconnection Networks [5]. A typical $N \times N$ multi-stage interconnection network uses $\log N$ stages of 2×2 crossbars with $N/2$ number of switches per stage ($O(N \log N)$ circuit complexity). Such multi-stage interconnection networks have significantly greater delay characteristics compared to the electrical crossbar and increased wiring complexity when scaling-up the network.

Free space optical crossbar switches have the potential to overcome the above limitations of multi-stage interconnection networks and electrical crossbars. Since directed light beams may non-destructively cross, free space optical crossbars overcome the circuit wiring complexities of the electrical counterpart. Furthermore, the number of input/output pins per substrate potentially may exceed 10^4, thus overcoming the restrictive pin-out limitations of electrical multi-stage interconnection networks and crossbars, and facilitating scalability of the system [17]. Several design approaches, such as beam stearing and beam spreading/masking have been proposed [15]. To date, technological limitations, such as

*The author is employed and supported by IBM Corp., Endicott, N.Y.

†Supported in part by the National Science Foundation Grant #CCR-9015717.

COMM'92-8/92/MD,USA

the switching speed of spatial light modulators, still inhibit implementation.

We believe that this technology will become feasible and that future networks could be comprised of asynchronous non-buffered interm switching nodes implemented by such optical crossbars. In such a system, a request to establish a route will contain the information encoded in its header that would specify which output to use at each intermediate switch in the interconnection network for the subsequent data transfer. The system operates in circuit-switching mode where all routing decisions are shifted to the periphery of the interconnection network [1].

Asynchronous, non-buffered switch operation may be required since light may not be buffered; light messages may be converted to electrical energy, and then buffered as such. This, however, introduces serious delay and thus we believe that future optical networks will not utilize optical-to-electrical conversion at intermediate network stages.

Data flow in such a network would consist of different traffic types (*i.e.,* voice, video, interactive data) each with different arrival and service statistics. Each traffic type may have different bandwidth requirements. It is of interest, therefore, to model and analyze such a network switch for multiple classes of traffic in order to quantify system performance and observe the effect of one traffic source on another.

In this paper, we model an $N_1 \times N_2$ crossbar interconnection network with bursty arrival statistics. In our previous papers [28, 29, 30], we analyzed the cases of $N_1 = N_2$ for simply one uniform arrival rate, non-uniform (hot spot) access patterns, and multiple Poisson arrival traffic, respectively. In this paper, we consider the more general case of multiple sources of uniform traffic and multiple arrival rates with bursty statistics. Furthermore, we assume that all input i to output j connections support the same bandwidth. If a particular class of traffic requires a greater bandwidth than that which is supported by one connection, several connections may be acquired to support the traffic class.

Since arrival traffic, in practice, may not necessarily be Poisson, it is of interest to study the performance characteristics of the model when it is subjected to bursty (non-Poisson) arrival statistics. It has been pointed out that peaky arrival traffic is well-approximated by the Pascal distribution [33]. Similarly, the Bernoulli distribution approximates smooth arrival traffic [24]. Due to the similarity of the Pascal and Bernoulli distributions, and since in the limiting case both distributions degenerate to the Poisson distribution, it is possible to consider a unified approximation of the smooth, regular and peaky arrival statistics [11]. The Bernoulli-Poisson-Pascal (BPP) distribution is a useful approximation of varied arrival traffic statistics.

An important characteristic of the proposed model is the assumption that connection requests are permitted to propagate into the crossbar switch fabric asynchronously (unslotted), as soon as the request arrives at the switch input. This is contrasted with the well known synchronous (slotted) crossbar model which has been suggested as an implementation of non-blocking ATM switches in fast packet switching networks. In a synchronous crossbar, arrival requests are permitted to propagate into the switch fabric with respect to a clock pulse or slot time [26]. Furthermore, in the context of today's multiprocessor systems, one assumes that contention does not exist at the switch inputs, since a processor does not initiate two or more concurrent requests. In the model of an asynchronous crossbar switch considered in this paper, however, switch interference can arise both from concurrent requests to the same switch input or the same output, making the model more difficult to analyze.

The main contribution of this paper is the study of bursty arrival statistics for an $N_1 \times N_2$ multi-rate crossbar switch model. The paper is organized as follows: The $N_1 \times N_2$ crossbar switch model is developed, the performance measures are derived and several efficient recursive algorithms to compute the performance measures are presented. A "revenue" oriented performance analysis is shown. Finally, some numerical examples are presented.

2 The Model

Consider an $N_1 \times N_2$ crossbar network [21, 26]. Assume that an input i can be connected to just one output j at a given instant. There are R types or classes of connection requests. Traffic of class $r \in R$ requests a_r inputs and outputs. Requests of type r for a particular set of inputs $(i_1, \ldots, i_{a_r})$ and a particular set of outputs $(j_1, \ldots, j_{a_r})$ arrive according to a discrete Markov process with rate $\lambda_r(k_r)$, where k_r is the number of concurrent connections of type r. Requests for a particular set of inputs $(i_1, \ldots, i_{a_r})$ and *any* set of outputs $(j_1, \ldots, j_{a_r})$ arrive with rate $\tilde{\lambda}_r(k_r)$, where $\tilde{\lambda}_r(k_r) = \binom{N_2}{a_r} \lambda_r(k_r)$. Assume a uniform traffic pattern. An established path

[1]This is somewhat analogous to *source-based routing* for some emerging high-speed packet-switching networks [8]. In these networks, packets contain complete routing information from the source. The intermediate nodes do not carry out any computations, just fast switching based on the information in the header. The computation of routing, flow control and end-to-end error control is the responsibility of the source and destination, not of the intermediate switching nodes. Such departure from traditional networks is motivated by a tremendous increase in the communication bandwidth and the low bit error rates due to reliable transmission mechanisms: the intermediate nodes should not perform any non-trivial computations so as not to become the performance bottlenecks.

of type r is used for a period of time distributed exponentially with mean $1/\mu_r$. This assumption will be relaxed to any distribution with mean $1/\mu_r$. In the crossbar switch under consideration, there are no buffers. Therefore, connection requests interfere if they try to access the same input or output. We assume that blocked requests are cleared from the system and that the recovery is managed by the corresponding end-points at the boundaries of the network.

Define the state of the system by the row vector $\mathbf{k} = (k_1, \ldots, k_R)$, where k_r is the number of concurrent connections of type r. Define $\mathbf{A} = (a_1, a_2, \ldots, a_R)$ as a column vector where a_r is the number of inputs required by traffic type r. Let $\mathbf{1}_r$ be the $1 \times R$ row vector with a 1 in position r, and zero elsewhere. In particular $\mathbf{k} \cdot \mathbf{A} = k_1 a_1 + \cdots + k_R a_R$. Define $\mathbf{N} = (N_1, N_2)$.

The state space of the system is then

$$\Gamma(\mathbf{N}) = \{\mathbf{k} \quad | \quad 0 \leq \mathbf{k} \cdot \mathbf{A} \leq \min(N_1, N_2)\}.$$

If the system is in state $\mathbf{k}$, then there are $(N_1 - \mathbf{k} \cdot \mathbf{A}) \times (N_2 - \mathbf{k} \cdot \mathbf{A})$ connection requests that do not interfere with any other connections already in progress. Therefore, if the system is in state $\mathbf{k}$, the probability intensity for a request of type r to be accepted is

$$q(\mathbf{k}, \mathbf{k} + \mathbf{1}_r) = (N_1 - \mathbf{k} \cdot \mathbf{A})(N_2 - \mathbf{k} \cdot \mathbf{A})\lambda_r(k_r),$$

and the probability intensity for a connection of type r to be finished is given by

$$q(\mathbf{k}, \mathbf{k} - \mathbf{1}_r) = k_r \mu_r.$$

Let us now define $\Psi(\cdot)$ and $\Phi_r(\cdot)$ by

$$\Psi(\mathbf{k}) = \frac{N_1!}{(N_1 - \mathbf{k} \cdot \mathbf{A})!} \cdot \frac{N_2!}{(N_2 - \mathbf{k} \cdot \mathbf{A})!},$$

and

$$\Phi_r(k_r) = \prod_{l=1}^{k_r} \frac{\lambda_r(l-1)}{l\mu_r}.$$

It is easy to verify that

$$\frac{q(\mathbf{k}, \mathbf{k} + \mathbf{1}_r)}{q(\mathbf{k} + \mathbf{1}_r, \mathbf{k})} = \frac{\Phi_r(k_r + 1)}{\Phi_r(k_r)} \cdot \frac{\Psi(\mathbf{k} + \mathbf{1}_r)}{\Psi(\mathbf{k})}. \tag{1}$$

It follows that if we define $\mathbf{k}(t)$ to be the state of the system at time t, then the underlying stochastic process $\{\mathbf{k}(t), t \geq 0\}$ is Markov. The ratio of Φ-functions reflects the ratio of birth and death rates for connection requests, whereas the ratio of Ψ-functions represents the modifying effect of resource availability. It can be shown that the underlying process is reversible ([19] theorem 1.3). In particular, we can show that the process $\mathbf{k}(t)$ has a unique steady-state probability distribution $\pi(\mathbf{k})$ of the following product form:

$$\begin{aligned}\pi(\mathbf{k}) &= \frac{1}{G(\mathbf{N})} \cdot \Psi(\mathbf{k}) \cdot \prod_{r=1}^{R} \Phi_r(k_r) \\ &= \frac{1}{G(\mathbf{N})} \cdot \frac{N_1!}{(N_1 - \mathbf{k} \cdot \mathbf{A})!} \cdot \frac{N_2!}{(N_2 - \mathbf{k} \cdot \mathbf{A})!} \\ &\quad \times \prod_{r=1}^{R} \left[\prod_{l=1}^{k_r} \frac{\lambda_r(l-1)}{l\mu_r}\right],\end{aligned} \tag{2}$$

where $G(\mathbf{N})$ is the normalization function given by[2]

$$\begin{aligned}G(\mathbf{N}) &= \sum_{\mathbf{k} \in \Gamma(\mathbf{N})} \frac{N_1!}{(N_1 - \mathbf{k} \cdot \mathbf{A})!} \cdot \frac{N_2!}{(N_2 - \mathbf{k} \cdot \mathbf{A})!} \\ &\quad \times \prod_{r=1}^{R} \left[\prod_{l=1}^{k_r} \frac{\lambda_r(l-1)}{l\mu_r}\right].\end{aligned} \tag{3}$$

It is easy to verify the detailed balance equation $\pi(\mathbf{k})q(\mathbf{k}, \mathbf{k} + \mathbf{1}_r) = \pi(\mathbf{k} + \mathbf{1}_r)q(\mathbf{k} + \mathbf{1}_r, \mathbf{k})$. It can be shown that the underlying stochastic process is insensitive; we can replace the exponential service distribution by any distribution with the same mean [7].

To model bursty traffic, we consider the Bernoulli-Poisson-Pascal (BPP) state-dependent arrival process of the form [11, 13]:

$$\lambda_r(k_r) = \alpha_r + \beta_r k_r,$$

where α and β are the statistical parameters characterizing the state-independent and state-dependent portions of the BPP process, respectively. This is termed Bernoulli-Poisson-Pascal since, depending on the parameters chosen, the distribution of the number of busy resources on an infinite server would be Bernoulli for $\beta_r < 0$, α_r/β_r a negative integer and $\beta_r \geq -\alpha_r/\max(N_1, N_2)$, Poisson for $\beta_r = 0$, or Pascal for $\alpha_r \geq 0$ and $0 < \beta_r < 1$. Note that in the Bernoulli case, α_r and β_r must be chosen so that $\alpha_r + \beta_r n \geq 0$ for $n \leq \max(N_1, N_2)$. As stated earlier, the BPP distribution is a unified approximation for peaky and smooth traffic statistics [11]. The mean M, variance V and peakedness Z (Z-factor) of the BPP distribution are given by

$$M = \frac{\alpha_r}{(1 - \beta_r)}, \quad V = \frac{\alpha_r}{(1 - \beta_r)^2}, \quad Z = \frac{V}{M} = \frac{1}{(1 - \beta_r)}.$$

Note that the Z-factor indicates the peakedness of the arrival traffic, either peaky ($Z > 1$), regular ($Z = 1$) or smooth ($Z < 1$). Let us now partition all request classes into two groups:

R_1 − The classes of requests with Poisson arrivals ($\beta_r = 0$).

[2] When $k_r = 0$, the product $[\prod_{l=1}^{k_r} \lambda_r(l-1)/l\mu_r]$ in equations 2 and 3 is defined to be 1.

R_2 – The classes of requests with Bernoulli or Pascal arrivals ($\beta_r \neq 0$).

Define $\rho_r = \alpha_r/\mu_r$ for $r \in R_1$ and $r \in R_2$. Earlier we defined $\lambda_r(k_r) = \tilde{\lambda}_r(k_r)/\binom{N_2}{a_r}$ so that $\rho_r = \tilde{\rho}_r/\binom{N_2}{a_r}$, $\alpha_r = \tilde{\alpha}_r/\binom{N_2}{a_r}$ and $\beta_r = \tilde{\beta}_r/\binom{N_2}{a_r}$. The probability distribution $\pi(\mathbf{k})$ can then be rewritten in the following form:

$$\begin{aligned}\pi(\mathbf{k}) &= \frac{1}{G(\mathbf{N})} \cdot \frac{N_1!}{(N_1 - \mathbf{k}\cdot\mathbf{A})!} \cdot \frac{N_2!}{(N_2 - \mathbf{k}\cdot\mathbf{A})!} \\ &\quad \times \prod_{r\in R_1} \frac{\rho_r^{k_r}}{k_r!} \cdot \prod_{r \in R_2} \left[\prod_{l=1}^{k_r} \frac{\beta_r(\alpha_r/\beta_r - 1 + l)}{\mu_r l}\right] \\ &= \frac{1}{G(\mathbf{N})} \cdot \frac{N_1!}{(N_1 - \mathbf{k}\cdot\mathbf{A})!} \cdot \frac{N_2!}{(N_2 - \mathbf{k}\cdot\mathbf{A})!} \\ &\quad \times \prod_{r\in R_1} \frac{\rho_r^{k_r}}{k_r!} \cdot \prod_{r \in R_2} \left[(\beta_r/\mu_r)^{k_r} \binom{\alpha_r/\beta_r - 1 + k_r}{k_r}\right].\end{aligned}$$

The model under consideration may also be interpreted in the following equivalent way: requests of class r arrive according to a Poisson process with unit rate but the service rate is of the state-dependent form:

$$\mu_r(k_r) = \frac{k_r \mu_r}{v_r + \delta_r k_r}$$

Note that the steady-state distribution in this case will be identical to that with BPP arrivals and state-independent service times when $\alpha_r = v_r + \delta_r$ and $\beta_r = \delta_r$. Such state-dependent service rates (with $v_r + \delta_r = 1$) have been considered in [16] within the context of a queueing system. Note that the case $\delta_r = 0$ corresponds to an infinite server node with Poisson arrivals ($\beta_r = 0$). In a queueing system [16], the case $\delta_r > 1$ can be used to model the slow-down due to congestion, whereas the case $0 < \delta_r < 1$ models the improvement of efficiency with congestion. We generalize this load dependence by omitting the restriction that $v_r + \delta_r = 1$ to obtain a general parameterized service. For example, if $\delta_r = 1$ and v_r is large, then $\mu_r(k_r)$ is linear with k_r for small k_r, and asymptotically approaches a constant for large k_r. For the purposes of this paper, we view the model in terms of the state-dependent arrival process rather than the state-dependent service process.

3 The Performance Measures

We now turn to the derivation of the performance measures of interest. We start by computing the concurrency (the average number of connections for each traffic type r where $r \in R_1$):

$$\begin{aligned}E_r(\mathbf{N}) &= \sum_{\mathbf{k}\in\Gamma(\mathbf{N})} k_r \pi(\mathbf{k}) = \frac{\rho_r}{G(\mathbf{N})} \cdot \frac{\partial G(\mathbf{N})}{\partial \rho_r} \\ &= \rho_r \binom{N_1}{a_r}\binom{N_2}{a_r} \cdot \frac{G(\mathbf{N} - a_r\mathbf{I})}{G(\mathbf{N})},\end{aligned}$$

where $\mathbf{I}$ is a unit row vector.

For the case in which $r \in R_2$, the average number of connections for each traffic type r is given by

$$\begin{aligned}E_r(\mathbf{N}) &= \sum_{\mathbf{k}\in\Gamma(\mathbf{N})} k_r \pi(\mathbf{k}) = \frac{\beta_r/\mu_r}{G(\mathbf{N})} \cdot \frac{\partial G(\mathbf{N})}{\partial(\beta_r/\mu_r)} \\ &= \frac{\binom{N_1}{a_r}\binom{N_2}{a_r}}{G(\mathbf{N})}\left\{\rho_r G(\mathbf{N} - a_r\mathbf{I}) + \frac{\beta_r}{\mu_r}\frac{\partial G(\mathbf{N} - a_r\mathbf{I})}{\partial(\beta_r/\mu_r)}\right\} \\ &= \binom{N_1}{a_r}\binom{N_2}{a_r}\frac{G(\mathbf{N} - a_r\mathbf{I})}{G(\mathbf{N})}\left\{\rho_r + \frac{\beta_r}{\mu_r}E_r(\mathbf{N} - a_r\mathbf{I})\right\}.\end{aligned}$$

The non-blocking probability is found by summing $\pi(\mathbf{k})$ over all states where a_r inputs and outputs are idle. It can easily be shown that the non-blocking probability is given by

$$B_r(\mathbf{N}) = \sum_{\mathbf{k}\in\Gamma(\mathbf{N}-a_r\mathbf{I})} k_r \pi(\mathbf{k}) = \frac{G(\mathbf{N} - a_r\mathbf{I})}{G(\mathbf{N})}. \qquad (4)$$

4 Revenue Oriented Performance Analysis

To provide an overall measure of the switch performance and to see the effects of load changes for some types of connection requests, we follow the approach used in [20] and suppose that an accepted connection of type r generates "revenue" w_r. The average return from the system is

$$W(\mathbf{N}) = \sum_{r\in R_1} w_r E_r(\mathbf{N}) + \sum_{r \in R_2} w_r E_r(\mathbf{N}).$$

If we write $w_r = \gamma_r \mu_r$, then it is clear that the total revenue ($W(\mathbf{N})$) is just the weighted throughput of the system with γ_r as weights. In particular, if $\gamma_1 = \cdots = \gamma_R = 1$, then the revenue is just the throughput of the system. These weights measure the relative significance that we attach to different types of connection requests.

To see the effect of load changes on the system performance, we compute the gradient of the weighted throughput with respect to ρ_r for Poisson arrival traffic ($r \in R_1$) or with respect to β_r/μ_r for bursty arrival traffic ($r \in R_2$). For the case of $R_2 = 0$, $R_1 \neq 0$, the gradient of the weighted throughput may be written as

$$\frac{\partial W(\mathbf{N})}{\partial \rho_r} = N_1 N_2 B_r(\mathbf{N})\Big(w_r - \Big[W(\mathbf{N}) - W(\mathbf{N} - a_r\mathbf{I})\Big]\Big).$$

From the above expression, the effect of increasing the load for connections of type r has the following economic interpretation: a request of type r is accepted with probability $B_r(\mathbf{N})$. If accepted, it will generate revenue w_r

directly, but at a cost of $\Delta W(\mathbf{N}) = W(\mathbf{N}) - W(\mathbf{N} - a_r\mathbf{I})$. In other words, $\Delta W(\mathbf{N})$ has the interpretation of a shadow cost. If $w_r > \Delta W(\mathbf{N})$ then the weighted throughput increases as we increase ρ_r. If, on the other hand, $w_r < \Delta W(\mathbf{N})$ then the weighted throughput decreases as we increase ρ_r. Even though ρ_r is higher, connections of type r would prevent connections of other types, resulting in the loss of revenue (*i.e.*, decrease in weighted throughput).

A closed form expression for the gradient of the weighted throughput was not found for the more general case of $R_2 \neq 0$, $R_1 \neq 0$. We use numerical methods to approximate the gradient $\partial W(\mathbf{N})/\partial(\beta_r/\mu_r)$ where $r \in R_2$ to show the effect of non-Poisson load changes on system performance. The gradient $\partial W(\mathbf{N})/\partial(\beta_r/\mu_r)$ is approximated via a forward difference.

5 Computation of the Performance Measures

Despite the simple expressions for the performance measures, to compute them exactly we must compute the normalization function. It is clear that a straightforward computation of $G(\mathbf{N})$ is impractical due to the factorial terms in equation 3. We therefore develop recursive algorithms to compute the performance measures.

To that end, define $Q(\mathbf{N}) = G(\mathbf{N})/N_1!N_2!$. Let $Z(\mathbf{t})$ be the generating[3] function of $Q(\mathbf{N})$ where $\mathbf{t} = (t_1, t_2)$. Using the identities

$$\begin{aligned}
\sum_{k=0}^{\infty}\binom{a-1+k}{k}y^k &= \left(\frac{1}{1-y}\right)^a, \\
\sum_{k=0}^{\infty}y^k &= \frac{1}{1-y}, \qquad 0<y<1, \\
\sum_{k=0}^{\infty}\frac{y^k}{k!} &= \exp y,
\end{aligned}$$

we have

$$\begin{aligned}
Z(\mathbf{t}) &= \sum_{\mathbf{N}=0}^{\infty} Q(\mathbf{N}) t_1^{N_1} t_2^{N_2} \\
&= \sum_{\mathbf{N}=0}^{\infty}\Bigg[\sum_{\mathbf{k}\in\Gamma(\mathbf{N})} \frac{t_1^{N_1-\mathbf{k}\,\mathbf{A}}}{(N_1-\mathbf{k}\cdot\mathbf{A})!}\cdot\frac{t_2^{N_2-\mathbf{k}\,\mathbf{A}}}{(N_2-\mathbf{k}\cdot\mathbf{A})!} \\
&\quad\times\left\{\prod_{r\in R_1}\frac{(\rho_r t_1^{a_r} t_2^{a_r})^{k_r}}{k_r!}\right\} \\
&\quad\times\left\{\prod_{r\in R_2}(\beta_r t_1^{a_r} t_2^{a_r}/\mu_r)^{k_r}\binom{\alpha_r/\beta_r-1+k_r}{k_r}\right\}\Bigg] \\
&= \exp\left(t_1+t_2+\sum_{r\in R_1}\rho_r t_1^{a_r} t_2^{a_r}\right) \\
&\quad\times\prod_{r\in R_2}\left(\frac{1}{1-(\beta_r t_1^{a_r} t_2^{a_r}/\mu_r)}\right)^{\alpha_r/\beta_r}. \qquad (5)
\end{aligned}$$

We define $\mathbf{t}^{\mathbf{N}} = t_1^{N_1} t_2^{N_2}$. Differentiating $Z(\mathbf{t})$ with respect to t_i we have

$$\begin{aligned}
\frac{\partial Z(\mathbf{t})}{\partial t_i} &= \sum_{\mathbf{N}=0}^{\infty}(N_i+1)Q(\mathbf{N}+\mathbf{1}_i)\mathbf{t}^{\mathbf{N}} \qquad (6) \\
&= Z(\mathbf{t})\Bigg[1+\sum_{r\in R_1}a_r\rho_r\mathbf{t}^{(a_r\mathbf{I}-\mathbf{1}_i)} \\
&\quad+\sum_{r\in R_2}a_r\rho_r\mathbf{t}^{(a_r\mathbf{I}-\mathbf{1}_i)}\left(\frac{1}{1-(\beta_r t_1^{a_r} t_2^{a_r}/\mu_r)}\right)\Bigg] \\
&= \sum_{\mathbf{N}=0}^{\infty}Q(\mathbf{N})\mathbf{t}^{\mathbf{N}}\Bigg[1+\sum_{r\in R_1}a_r\rho_r\mathbf{t}^{(a_r\mathbf{I}-\mathbf{1}_i)} \\
&\quad+\sum_{r\in R_2}a_r\rho_r\mathbf{t}^{(a_r\mathbf{I}-\mathbf{1}_i)}\left(\sum_{m=0}^{\infty}\left(\frac{\beta_r t_1^{a_r} t_2^{a_r}}{\mu_r}\right)^m\right)\Bigg], \qquad (7)
\end{aligned}$$

where vector $\mathbf{1}_i$ is defined as $\mathbf{1}_1 = (1,0)$ or $\mathbf{1}_2 = (0,1)$, respectively.

Equating the corresponding powers of $t_1^N t_2^N$ we obtain from equations 6 and 7

$$\begin{aligned}
Q(\mathbf{N}+\mathbf{1}_i) &= \frac{Q(\mathbf{N})}{N_i+1}+\sum_{r\in R_1}a_r\rho_r\frac{Q(\mathbf{N}+\mathbf{1}_i-a_r\mathbf{I})}{N_i+1} \\
&\quad+\sum_{r\in R_2}a_r\rho_r\Bigg[\sum_{m=0}^{\min(N_1,N_2)}\left(\frac{\beta_r}{\mu_r}\right)^m \\
&\quad\times\frac{Q(\mathbf{N}+\mathbf{1}_i-(m+1)a_r\mathbf{I})}{N_i+1}\Bigg]. \qquad (8)
\end{aligned}$$

Let us define $V(\mathbf{N}+\mathbf{1}_i, r)$ for $r \in R_2$ as

$$\begin{aligned}
V(\mathbf{N}+\mathbf{1}_i,r) &= \sum_{m=0}^{\min(N_1,N_2)}\Bigg\{\left(\frac{\beta_r}{\mu_r}\right)^m \\
&\quad\times Q(\mathbf{N}+\mathbf{1}_i-(m+1)a_r\mathbf{I})\Bigg\} \\
&= Q(\mathbf{N}+\mathbf{1}_i-a_r\mathbf{I}) \\
&\quad+\left(\frac{\beta_r}{\mu_r}\right)V(\mathbf{N}+\mathbf{1}_i-a_r\mathbf{I},r). \qquad (9)
\end{aligned}$$

Therefore, the recurrence relation in equation 8 may be re-written as

$$\begin{aligned}
Q(\mathbf{N}+\mathbf{1}_i) &= \frac{Q(\mathbf{N})}{N_i+1}+\sum_{r\in R_1}a_r\rho_r\frac{Q(\mathbf{N}+\mathbf{1}_i-a_r\mathbf{I})}{N_i+1} \\
&\quad+\sum_{r\in R_2}a_r\rho_r\frac{V(\mathbf{N}+\mathbf{1}_i,r)}{N_i+1}. \qquad (10)
\end{aligned}$$

[3] *i.e.*, the exponential generating function of $G(\mathbf{N})$.

The third term on the right hand side of equation 10 may be computed recursively using equation 9. Given the following definition of permutations,

$$P(N_i, a_r) = \frac{N_i!}{(N_i - a_r)!}, \tag{11}$$

we obtain the following algorithm for computing the performance measures:

Algorithm 1

Step 1. *Initialize* $\mathbf{n} = (n_1, n_2) = \mathbf{0}$, $Q(\mathbf{0}) = 1$
Step 2. *For* $\mathbf{0} < \mathbf{n} < \mathbf{N}$ *compute:*

$$\begin{aligned} Q(\mathbf{n} + \mathbf{1}_i) &= \frac{Q(\mathbf{n})}{n_i + 1} + \sum_{r \in R_1} a_r \rho_r \frac{Q(\mathbf{n} + \mathbf{1}_i - a_r \mathbf{I})}{n_i + 1} \\ &+ \sum_{r \in R_2} a_r \rho_r \frac{V(\mathbf{n} + \mathbf{1}_i, r)}{n_i + 1} \end{aligned}$$

Step 3. *Compute the performance measures:*

$$\begin{aligned} B_r(\mathbf{N}) &= \frac{Q(\mathbf{N} - a_r \mathbf{I})}{P(N_1, a_r) P(N_2, a_r) Q(\mathbf{N})} \\ E_r(\mathbf{N}) &= \frac{\rho_r Q(\mathbf{N} - a_r \mathbf{I})}{P(N_1 - 1, a_r) P(N_2 - 1, a_r) Q(\mathbf{N})}, r \in R_1 \\ E_r(\mathbf{N}) &= \frac{Q(\mathbf{N} - a_r \mathbf{I})}{P(N_1 - 1, a_r) P(N_2 - 1, a_r) Q(\mathbf{N})} \\ &\times \left\{ \rho_r + \frac{\beta_r}{\mu_r} E_r(\mathbf{N} - a_r \mathbf{I}) \right\}, \quad r \in R_2 \end{aligned}$$

where $E_r(\mathbf{0}) = \mathbf{0}$.

Let us now evaluate the complexity of Algorithm 1. For each successive step of the recurrence, the first term on the right hand side of equation 10 requires $O(1)$ operations, the second term requires $O(R_1)$ operations and the third term requires $O(R_2)$ operations. Since $N_1 N_2$ iterations are required to compute $Q(N_1, N_2)$, the complexity of Algorithm 1 is $O(N_1 N_2 (R_1 + R_2))$.

5.1 Mean Value Analysis

We now present a mean-value type of algorithm to compute the blocking probabilities. This second algorithm is cast directly in terms of blocking probabilities. The main advantage of this approach is numerical stability[4]. The derivation presented here is analogous to the generalized approach presented in [27].

Define $F_i(\mathbf{N})$ as a ratio of the normalization functions.

$$F_i(\mathbf{N}) = \frac{Q(\mathbf{N} - \mathbf{1}_i)}{Q(\mathbf{N})}. \tag{12}$$

[4]Dynamic scaling could also be used as suggested in [10] and discussed in Section 6.

Also define

$$\begin{aligned} H_r(\mathbf{N}) &= \frac{Q(\mathbf{N} - a_r \mathbf{I})}{Q(\mathbf{N})} \\ &= \prod_{j=1}^{2} \prod_{m=1}^{a_r} F_j\left(\mathbf{N} - a_r \mathbf{I} + (j-1) a_r \mathbf{1}_{j-1} + m \mathbf{1}_j\right). \end{aligned} \tag{13}$$

Note that $H_r(\mathbf{N})$ may be rewritten for any $1 \le j \le 2$ as

$$H_r(\mathbf{N}) = F_j(\mathbf{N}) L_{jr}(\mathbf{N} - \mathbf{1}_j), \tag{14}$$

where

$$\begin{aligned} L_{jr}(\mathbf{N} - \mathbf{1}_j) &= \prod_{k=1}^{a_r - 1} F_j(\mathbf{N} - a_r \mathbf{1}_j + k \mathbf{1}_j) \\ &\times \prod_{\substack{m=1 \\ i \neq j}}^{a_r} F_i(\mathbf{N} - a_r \mathbf{I} + m \mathbf{1}_i). \end{aligned} \tag{15}$$

We are now in a position to derive the proposed mean-value algorithm. Dividing both sides of equation 8 by $Q(\mathbf{N} + \mathbf{1}_i)$ and multiplying by $N_i + 1$, we obtain

$$\begin{aligned} N_i + 1 &= \frac{Q(\mathbf{N})}{Q(\mathbf{N} + \mathbf{1}_i)} + \sum_{r \in R_1} a_r \rho_r \frac{Q(\mathbf{N} + \mathbf{1}_i - a_r \mathbf{I})}{Q(\mathbf{N} + \mathbf{1}_i)} \\ &+ \sum_{r \in R_2} a_r \rho_r \left[\sum_{m=0}^{\min(N_1, N_2)} \left(\frac{\beta_r}{\mu_r} \right)^m \right. \\ &\times \left. \frac{Q(\mathbf{N} + \mathbf{1}_i - (m+1) a_r \mathbf{I})}{Q(\mathbf{N} + \mathbf{1}_i)} \right]. \end{aligned} \tag{16}$$

Let us now define for the class $r \in R_2$

$$D(r, \mathbf{N}) = \sum_{m=0}^{\min(\mathbf{N})} \left(\frac{\beta_r}{\mu_r} \right)^m \frac{Q(\mathbf{N} - m a_r \mathbf{I})}{Q(\mathbf{N})}. \tag{17}$$

Given the above definitions, we may rewrite equation 16 as

$$\begin{aligned} (N_i + 1) &= F_i(\mathbf{N} + \mathbf{1}_i) \\ &+ \sum_{r \in R_1} \rho_r a_r F_j(\mathbf{N} + \mathbf{1}_i) L_{jr}(\mathbf{N} + \mathbf{1}_i - \mathbf{1}_j) \\ &+ \sum_{r \in R_2} \rho_r a_r \Big[F_j(\mathbf{N} + \mathbf{1}_i) L_{jr}(\mathbf{N} + \mathbf{1}_i - \mathbf{1}_j) \\ &\times D(r, \mathbf{N} + \mathbf{1}_i - a_r \mathbf{I}) \Big]. \end{aligned} \tag{18}$$

Manipulating equation 17, it may be shown that

$$D(r, \mathbf{N}) = H_r(\mathbf{N}) + \frac{\beta_r}{\mu_r} D(r, \mathbf{N} - a_r \mathbf{I}). \tag{19}$$

Using equations 12, 13 and 14, it may be shown that

$$L_{jr}(\mathbf{N} - \mathbf{1}_j) = \frac{H_r(\mathbf{N} - \mathbf{1}_j)}{F_j(\mathbf{N} - a_r \mathbf{I})}. \tag{20}$$

Equations 14, 18, 19 and 20, then form the basis of the following mean-value algorithm.

Algorithm 2

Step 1. *For* $1 \le i \le 2, \quad 1 \le r \le R$*:*

$F_i(\mathbf{0}) = 0,\ F_1(n_1, 0) = n_1$ *for* $n_1 \ge 1$

$F_2(0, n_2) = n_2$ *for* $n_2 \ge 1$

$F_i(\mathbf{1}) = 1/(1 + \sum_{r \in R_1} a_r\rho_r + \sum_{r \in R_2} a_r\rho_r)$

$L_{ir}(\mathbf{0}) = 0,\ L_{ir}(\mathbf{1}_i) = 1$

$L_{ir}(\mathbf{1}) = 1/(1 + \sum_{r \in R_1} a_r\rho_r + \sum_{r \in R_2} a_r\rho_r)$

$L_{1r}(0, n_2) = n_2$ *for* $n_2 \ge 2$

$L_{1r}(n_1, 0) = n_1$ *for* $n_1 \ge 1$

$H_r(\mathbf{0}) = 0,\ H_r(\mathbf{1}_i) = H_r(\mathbf{1}) = 1,\ D(r, 0) = 0$

For each $r \ge 2$*:*

$L_{ir}(\mathbf{1}) = 0,\ L_{1r}(1,2) = 2,\ L_{2r}(2,1) = 2$

$L_{1r}(1, n_2) = L_{1r}(1, n_2 - 1) \cdot n_2$ *for* $n_2 > 2$

$L_{2r}(n_1, 1) = L_{2r}(n_1 - 1, 1) \cdot n_1$ *for* $n_1 > 2$

$F_i(\mathbf{0}) = 1$

Step 2. *For* $\mathbf{1} < \mathbf{n} < \mathbf{N}$*:*

For $1 \le i \le 2,\ 1 \le j \le 2$ *and* $i \ne j$*, compute:*

$$F_i(\mathbf{n} + \mathbf{1}_i) = (n_i + 1)/ \left[1 + \sum_{r \in R_1} a_r\rho_r L_{ir}(\mathbf{n}) + \sum_{r \in R_2} a_r\rho_r L_{ir}(\mathbf{n}) D(r, \mathbf{n} + \mathbf{1}_i - a_r\mathbf{I})\right]$$

$$F_j(\mathbf{n} + \mathbf{1}_i) = (n_i + 1) - F_i(\mathbf{n} + \mathbf{1}_i)/ \left[\sum_{r \in R_1} a_r\rho_r L_{jr}(\mathbf{n} + \mathbf{1}_i - \mathbf{1}_j) + \sum_{r \in R_2} a_r\rho_r L_{jr}(\mathbf{n} + \mathbf{1}_i - \mathbf{1}_j) D(r, \mathbf{n} + \mathbf{1}_i - a_r\mathbf{I})\right]$$

For $1 \le r \le R$*, compute* $H_r(\mathbf{n} + \mathbf{1}_i)$ *using equation 14.*

For $1 \le r \le R$*, compute* $L_{jr}(\mathbf{n} + \mathbf{1}_i)$ *and* $L_{ir}(\mathbf{n} + \mathbf{1}_i)$ *using equation 20.*

For $1 \le r \le R$ *compute* $D(r, \mathbf{n} + \mathbf{1}_i)$ *using equations 19.*

Step 3. *For* $1 \le r \le R$*: Compute the remaining performance measures:*

$$\begin{aligned}
B_r(\mathbf{N}) &= \frac{H_r(\mathbf{N})}{P(N_1, a_r)P(N_2, a_r)} \\
E_r(\mathbf{N}) &= \rho_r N_1 N_2 \cdot B_r(\mathbf{N}), \quad r \in R_1 \\
E_r(\mathbf{N}) &= N_1 N_2 B_r(\mathbf{N}) \Big\{\rho_r + \frac{\beta_r}{\mu_r} E_r(\mathbf{N} - a_r\mathbf{I})\Big\},
\end{aligned}$$

where $E_r(\mathbf{0}) = \mathbf{0}$ *and* $r \in R_2$*.*

Let us evaluate the complexity of Algorithm 2 . The main loop in step 2 requires $O(N_1N_2)$ iterations. The computation of each successive $F_i(\mathbf{n}+\mathbf{1}_i)$ value requires $O(R_1 + R_2)$ operations. Also, the computation of successive values of $L_{jr}(\mathbf{n}+\mathbf{1}_i)$, $H_r(\mathbf{n}+\mathbf{1}_i)$ and $D_r(\mathbf{n}+\mathbf{1}_i)$ for $1 \le r \le (R_1 + R_2)$ can be computed in $O(R_1 + R_2)$. Thus, the computational requirements for Algorithm 2 are $O((R_1 + R_2)N_1N_2)$. Algorithm 2 is preferable compared to Algorithm 1 because of its superior numerical stability. Algorithm 2, however, requires substantially more space in practice compared to Algorithm 1. Thus, Algorithm 1 is preferable for computing the performance measures of small dimension crossbars ($N \le 32$) whereas Algorithm 2 is advantageous for larger system sizes.

6 Dynamic Scaling

During the computation of $Q(\mathbf{N})$ in Algorithm 1, underflow or overflow may occur. Dynamic scaling can be used to compute the recurrence by introducing a scaling factor ω [9]. We may rewrite equation 10 as follows:

$$\begin{aligned}
\omega Q(\mathbf{N} + \mathbf{1}_i) &= \frac{\omega Q(\mathbf{N})}{N_i + 1} + \sum_{r \in R_1} a_r\rho_r \frac{\omega Q(\mathbf{N} + \mathbf{1}_i - a_r\mathbf{I})}{N_i + 1} \\
&+ \sum_{r \in R_2} a_r\rho_r \frac{\omega V(\mathbf{N} + \mathbf{1}_i, r)}{N_i + 1}.
\end{aligned}$$

The scaling factor is used to prevent underflow or overflow. If it is determined that the subsequent computation of $Q(\mathbf{n})$ will underflow or overflow, then we may scale equation 10 as shown above so that the problem is avoided. The scaling operation increases the complexity of Algorithm 1 by a constant factor and may be carried out at any step of the algorithm. Since the performance measures are expressed in terms of the ratio $Q(\mathbf{N} - a_r\mathbf{I})/Q(\mathbf{N})$, the scaling factor does not affect the performance measure results.

7 Results

We now show some numerical examples to illustrate the performance characteristics of the model. Figure 1 shows the blocking probability for $N_1 \times N_2$ crossbar systems, where $N_1 = N_2 = N$, given one arrival traffic type R_2 ($R_1 = 0$), and bandwidth requirement of one connection per arrival ($a_r = 1$). The arrival traffic parameters are such that $\tilde{\alpha}_r = .0024$, $\mu_r = 1.0$ and $\tilde{\beta}_r$ ranges from 0.0 to -4.0×10^{-6}. The $\tilde{\alpha}_r$, $\tilde{\beta}_r$ parameters are chosen to drive the non-blocking probability to approximately 99.5%, which may be considered an acceptable operating point (blocking probability $\approx$ 0.5%). Figure 1 corresponds to the Bernoulli approximation of smooth arrival traffic. Note that α_r/β_r is a negative integer, $\beta_r \le 0$ and $\alpha_r + \beta_r n \ge 0$ where $n = \max(N_1, N_2)$. The largest system size used in Figure 1 is 128 $\times$ 128 (n = 128). The solid curve shows the degenerate case (Poisson arrivals) when $\tilde{\beta}_r = 0$. We observe that the degenerate case provides an upper bound for the smooth arrival

Table 1: Total Load Calculations

N_1	$\tilde{\rho}_1$	$\tilde{\rho}_2$
4	.000600	.000800
8	.000300	.000171
16	.000150	.0000400
32	.0000750	.00000967
64	.0000375	.00000238

Table 1: Input parameters used to model two traffic types with bandwidth requirements $a_1 = 1$ and $a_2 = 2$

traffic. Furthermore, we found that the smooth arrival traffic affected the blocking probability proportionally at other operating points. For example, it is noted that for $N_1 = N_2 = 128$, the difference in blocking probabilities between the degenerate Poisson case ($\tilde{\alpha}_r = .0024$, $\tilde{\beta}_r = 0$) and the case of $\tilde{\alpha}_r = .0024$, $\tilde{\beta}_r = -4.0 \times 10^{-6}$ is approximately 0.1%. We found that at other operating points, the difference in blocking probabilities for the proportionally scaled $\tilde{\alpha}_r$ and $\tilde{\beta}_r$ was again 0.1%.

Figure 2 shows the blocking probability for system sizes $1 \leq N \leq 128$ ($N_1 = N_2 = N$) for peaky arrival traffic ($R_1 = 0$, $R_2 = 1$), and bandwidth requirement of one connection per arrival ($a_r = 1$). The solid curve shows the degenerate case (Poisson) where $\tilde{\alpha}_r = .0024$, $\mu_r = 1.0$ and $\tilde{\beta}_r = 0$. It is observed that peaky arrival traffic has a dramatic impact on blocking probability.

Figure 3 shows a comparison of the asynchronous crossbar blocking probabilities for two classes of arrival traffic ($R_1 = 1$, $R_2 = 1$) versus one class ($R_1 = 0$, $R_2 = 1$), and bandwidth requirement of one connection per arrival ($a_r = 1$). We observe that the class R_1 type traffic simply shifts the operating point of the crossbar. Also, the amount of $\tilde{\beta}_r$ (in class R_2) causes the same percentage change in blocking probability regardless of operating point ($\tilde{\alpha}_r$ from R_1, R_2). We observe from Figure 1, Figure 2 and Figure 3 that the blocking probability is relatively insensitive to moderate variations in the statistical parameters. This was observed for large ranges of statistical parameters.

Table 2 shows the revenue oriented performance results for two arrival traffic types (traffic $1 \in R_1$ is Poisson and traffic $2 \in R_2$ is bursty) with different weights, and bandwidth requirement of $a_r = 1$. In Table 2, the first set of parameters are $\tilde{\rho}_1 = .0012$, $\tilde{\rho}_2 = .0012$, $\tilde{\beta}_2 = .0012$, $w_1 = 1.0$ and $w_2 = .0001$. The traffic types are weighted unequally so that traffic type 1 returns a significantly higher revenue than traffic type 2. The second set of parameters in Table 2 are adjusted so that $\tilde{\beta}_2$ is increased from $\tilde{\beta}_2 = .0012$ to $\tilde{\beta}_2 = .0036$, thus indicating a small increase in the peakedness or Z-factor of traffic 2. All other parameters remain the same. The third set of parameters are adjusted so that $\tilde{\rho}_2$ is increased from $\tilde{\rho}_2 = .0012$ to $\tilde{\rho}_2 = .0036$, resulting in a large increase in the operating point of the crossbar. Since the gradient $\partial W / \partial(\beta_2/\mu_2)$ is negative, the overall weighted throughput ($W(\mathbf{N})$) decreases as load $\tilde{\beta}_2/\mu_2$ is increased, resulting in a loss of revenue. These results are consistent with the results presented in Figures 2 and 3 which showed that an increase in bursty traffic impacts the blocking probability, and hence the system throughput. It is observed that increasing $\tilde{\alpha}_2$ causes a greater decrease in revenue (and increase in non-blocking probability) compared to that resulting from the proportional increase in $\tilde{\beta}_2$.

Figure 4 shows the effect of multi-rate traffic on the crossbar blocking probability. Two traffic types are compared, both of which are Poisson ($r \in R_1$). Traffic $\tilde{\rho}_1$ has the bandwidth requirement of one crossbar connection per arrival ($a_r = 1$). Traffic $\tilde{\rho}_2$ has the bandwidth requirement of two crossbar connections per arrival ($a_r = 2$). The total load τ_r is kept constant at $\tau_r = .0048$ for all system sizes. The arrival rates $\tilde{\rho}_1$ and $\tilde{\rho}_2$ are calculated using $\tilde{\rho}_r = \tau_r / \binom{N_1}{a_r}$. The blocking probability is calculated analytically using the model, considering each traffic type separately. Table 1 shows the input parameters used for the analytical model. Figure 4 shows a comparison of the independent effect of the two traffic types on blocking probability. We observe that traffic $\tilde{\rho}_2$ with $a_2 = 2$ results in a significantly higher blocking probability as compared to traffic $\tilde{\rho}_1$ with $a_1 = 1$, for a constant overall crossbar load τ_r. This is due to the higher contention of two connection requests per arrival event for $\tilde{\rho}_2$ traffic as compared to one connection request per arrival event for $\tilde{\rho}_1$ traffic.

8 Conclusion

In this paper, we presented a model of an $N_1 \times N_2$ multi-rate crossbar switch for several arrival traffic distributions. The effects of bursty traffic on the performance of the asynchronous crossbar switch was quantified analytically. Recurrence relations to exactly compute the concurrency for each traffic type, and the non-blocking probability were derived. A revenue oriented approach to performance analysis was presented. The model could be applied to the performance analysis of all-optical circuit-switching networks which support various integrated multi-rate traffic types. Future work includes extending this analysis to asynchronous all-optical multi-stage networks, comparing our analytical results with simulation and analyzing other asynchronous switches.

References

[1] K. Batcher. Sorting Networks and their Applications. In *AFIPS Proc. of SJCC*, pages 307–314, vol. 32, 1968.

[2] V. Beneš. *Mathematical Theory of Connecting Networks and Telephone Traffic.* Academic Press, New York, 1965.

[3] D. Bhandarkar. Analysis of Memory Interference in Multiprocessors. *IEEE Trans. Computers*, 24(9):897–908, September 1975.

[4] D. Bhandarkar. Some Performance Issues in Multiprocessor System Design. *IEEE Trans. Computers*, 26(5):506–511, May 1977.

[5] L. Bhuyan and C. Lee. An Interference Analysis of Interconnection Networks. In *Proc. Int. Conf. on Parallel Processing*, pages 2–9, August 23-26th 1983.

[6] G. Broomell and J. Heath. An Integrated Circuit Crossbar Switching System. In *Proc. 4-th Int. Conf. on Distributed Computing Systems*, pages 278–287, May 1984.

[7] D. Burman, J. Lehoczky, and Y.Lim. Insensitivity of Blocking Probabilities in a Circuit-Switched Network. *J. Appl. Prob.*, 21:850–859, 1984.

[8] I. Cidon and I. Gopal. PARIS: An Approach to Integrated High Speed Networks. *Intern. J. Digital and Analog Cabled Systems*, 1:77–85, 1988.

[9] A. Conway and N. Georganas. A New Method for Computing the Normalization Constant of Multiple-Chain Queueing Networks. *INFOR.*, 24(3):184–198, 1986.

[10] A. Conway and N. Georganas. *Queueing Networks-Exact Computational Algorithms: A Unified Theory Based on Decomposition and Aggregation.* MIT Press, Cambridge, MA, 1989.

[11] L. Delbrouck. A Unified Approximate Evaluation of Congestion Functions for Smooth and Peaky Traffic. *IEEE. Trans. Commun.*, 29(2):85–91, February 1981.

[12] D. Diaz and J. Jump. Analysis and Simulation of Buffered Delta Networks. *IEEE Trans. Computers*, 30(4):273–282, April 1981.

[13] Z. Dziong and J. Roberts. Congestion Probabilities in a Circuit-Switched Integrated Services Network. *Performance Evaluation*, 7(4):267–284, November 1987.

[14] R. Goke and G. Lipovski. Banyan networks for Partitioning on Multiprocessor Systems. In *Proc. First Conf. on Computer Architecture*, pages 21–28, 1973.

[15] A. Hartmann and S. Redfield. Design Sketches for Optical Crossbar Switches Intended for Large-Scale Parallel Processing Applications. *Optical Engineering*, 28(4):315–327, April 1989.

[16] H. Heffes. Moment Formulae for a Class of Mixed Multi-Job-Type Queueing Networks. *Bell Syst. Tech. J.*, 61(5):709–745, June 1982.

[17] H. Hinton. Free-Space Systems and Devices. In *Proc. IEEE Int. Conf. on Communications.*, volume 3, pages 1146–1151, Atlanta, GA, April 16-19 1990.

[18] A. Huang and S. Knauer. Starlite: A Wideband Digital Switch. In *Proc. IEEE INFOCOM*, pages 121–125, 1984.

[19] F. Kelly. *Reversibility and Stochastic Networks.* J. Wiley and Sons, New York, 1979.

[20] F. Kelly. Routing in Circuit-Switched Networks: Optimization, Shadow Prices and Decentralization. *Adv. Appl. Prob.*, 20:112–144, March 1988.

[21] C. Kruskal and M. Snir. The Performance of Multistage Interconnection Networks for Multiprocessors. *IEEE Trans. Computers*, 32(12):1091–1098, December 1983.

[22] T. Lang and H. Stone. A Shuffle-Exchange Network with Simplified Control. *IEEE Trans. Computers*, 25:55–66, January 1976.

[23] M. Marsan, G. Balbo, G. Conte, and F. Gregoretti. Modeling Bus Contention and Memory Interference in a Multiprocessor System. *IEEE Trans. Computers*, 32(1):60–72, January 1983.

[24] R. Mina. Modification of the Input Process used in Classical Traffic Theory. In *Proc. 6th Int. Teletraffic Cong.*, Munich, West Germany, 1970.

[25] Y. Oie, T. Suda, M. Murata, and H. Miyahara. Survey of the Performance of Non-Blocking Switches with FIFO Buffers. In *Proc. IEEE Int. Conf. on Communications*, pages 316.1.1–316.1.5, April 1990.

[26] J. Patel. Performance of Processor-Memory Interconnections for Multiprocessors. *IEEE Trans. Computers*, 30(10):771–780, October 1981.

[27] E. Pinsky and A. Conway. Mean Value Analysis of Multi-Facility Models with State-Dependent Arrivals. 1991. Submitted for publication.

[28] E. Pinsky and P. Stirpe. Modeling and Analysis of Hot Spots in an Asynchronous $N \times N$ Crossbar Switch. In *Int. Conf. on Parallel Processing*, pages I–546–549, Chicago, Ill., August 1991.

[29] P. Stirpe and E. Pinksy. Performance Analysis of an Asynchronous Multirate Crossbar. In *Int. Symposium on Communications*, pages 558–561, Taiwan, Republic of China, 1991.

[30] P. Stirpe and E. Pinsky. The Performance Analysis of an Asynchronous Non-blocking Network Switch. In *Singapore Int. Conf. on Networks*, pages 7–12, Raffles City Convention Centre, Singapore, September 1991.

[31] H. Stone. Parallel Processing with the Perfect Shuffle. *IEEE Trans. Computers*, 20(2):153–161, February 1971.

[32] S. Thanawastien and V. Nelson. Interference Analysis of Shuffle/Exchange Networks. *IEEE Trans. Computers*, 30(8):545–556, August 1981.

[33] R. Wilkinson. Theories of Toll Traffic Engineering in the U.S.A. *Bell Syst. Tech. J.*, 35(2):421–514, 1956.

[34] Y. Yeh, M. Hluchyj, and A. Acampora. The Knockout Switch: A Simple Modular Architecture for High-Performance Packet Switching. *IEEE J. SEC*, SAC-5:1274–1283, 1987.

Table 2: Revenue Oriented Analysis

parameters	N	$\partial W/\partial \rho_1$	$\partial W/\partial(\beta_2/\mu_2)$	B_r(N)	W(N)
$\tilde{\rho}_1 = .0012$	1	0.99	–	0.00239425	0.00119725
$\tilde{\rho}_2 = .0012$	2	3.97	+2.38871e-07	0.00358566	0.00239163
$\tilde{\beta}_2 = .0012$	4	15.89	−2.12995e-05	0.00418083	0.00478041
$w_1 = 1.0$	8	63.57	−0.000370081	0.0044820	0.00955794
$w_2 = .0001$	16	254.22	−0.00402453	0.00464093	0.0191128
	32	1016.76	−0.0369292	0.00473733	0.0382221
	64	4066.62	−0.313413	0.0048195	0.0764381
	128	16264.50	−2.53805	0.00492849	0.152861
	256	65045.30	−19.3138	0.00511868	0.305671
$\tilde{\rho}_1 = .0012$	1	0.99	–	0.00239425	0.00119725
$\tilde{\rho}_2 = .0012$	2	3.97	+2.38871e-07	0.00358566	0.00239163
$\tilde{\beta}_2 = .0036$	4	15.89	−2.12788e-05	0.00418403	0.0047804
$w_1 = 1.0$	8	63.56	−0.00036904	0.00449504	0.00955782
$w_2 = .0001$	16	254.21	−0.00399684	0.00467581	0.0191122
	32	1016.68	−0.0363166	0.00481708	0.0382193
	64	4065.93	−0.299452	0.00498953	0.0764266
	128	16258.80	−2.09857	0.00527912	0.152817
	256	64998.30	−68.6054	0.00582948	0.305646
$\tilde{\rho}_1 = .0012$	1	0.99	–	0.00477707	0.00119463
$\tilde{\rho}_2 = .0036$	2	3.96	+7.13145e-07	0.00714287	0.00238357
$\tilde{\beta}_2 = .0012$	4	15.83	−6.30503e-05	0.0083221	0.00476149
$w_1 = 1.0$	8	63.28	−0.00109351	0.0089218	0.00951723
$w_2 = .0001$	16	253.05	−0.0118788	0.00924611	0.0190283
	32	1011.95	−0.108917	0.00945823	0.0380486
	64	4046.89	−0.923616	0.0096644	0.0760824
	128	16182.50	−7.47015	0.0099675	0.152123
	256	64693.50	−56.7188	0.010518	0.304099

Table 2: Loss in total revenue ($W(\mathbf{N})$) as traffic type 2 is increased

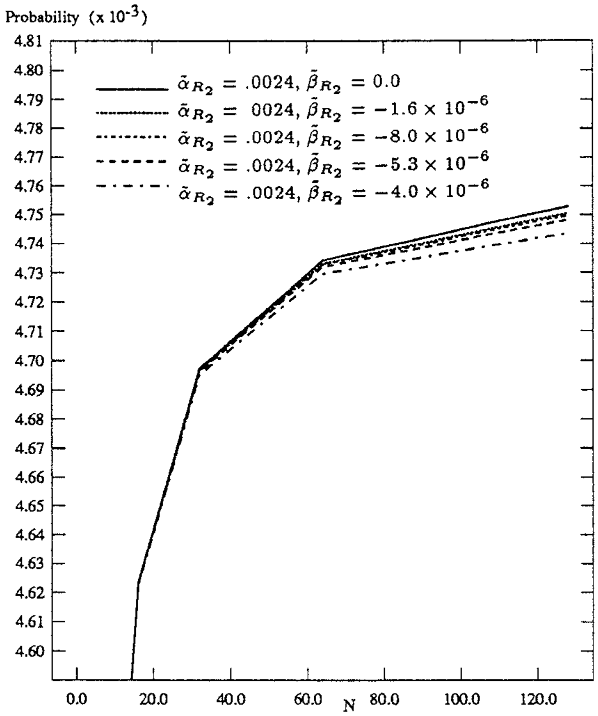

Figure 1: Smooth arrival traffic for the case of one type of class $R_2 = 1$ traffic, no traffic of type $R_1 = 0$ (Poisson) and $a_r = 1$

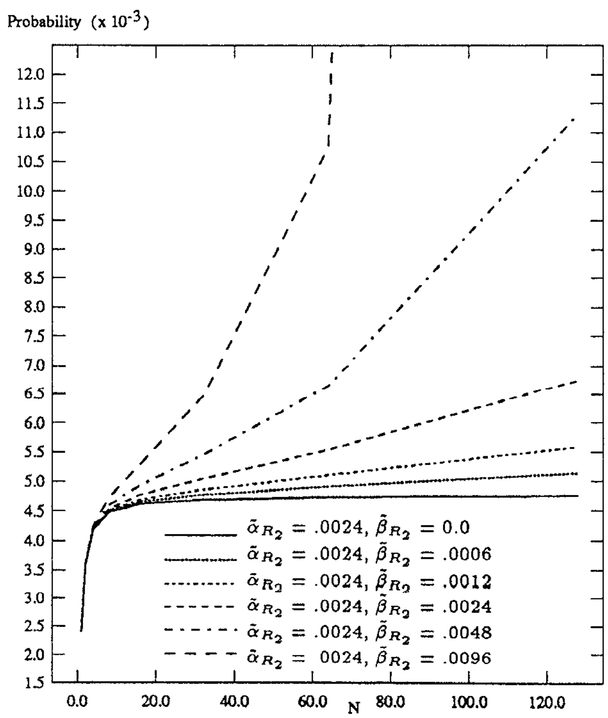

Figure 2: Peaky arrival traffic for the case of one type of class $R_2 = 1$ traffic, no traffic of type $R_1 = 0$ (Poisson) and $a_r = 1$

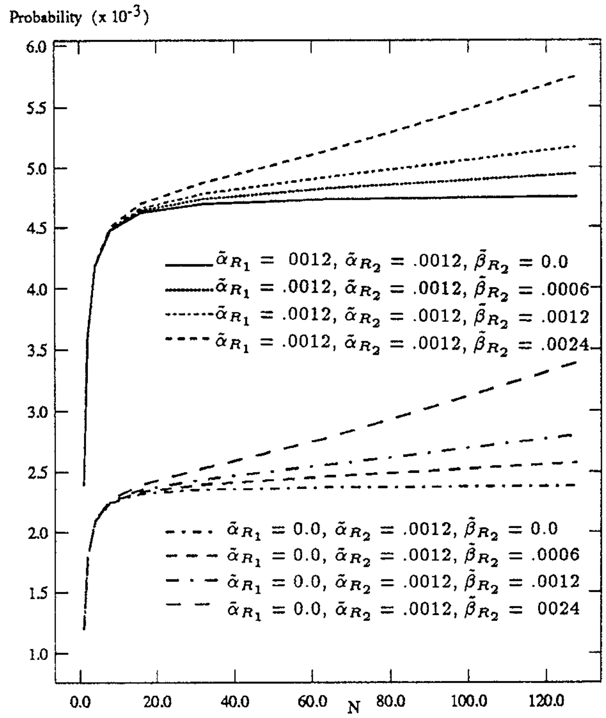

Figure 3: Peaky arrival traffic for the case of $R_1 = 1$, $R_2 = 1$ compared with $R_1 = 0$, $R_2 = 1$ for $a_r = 1$

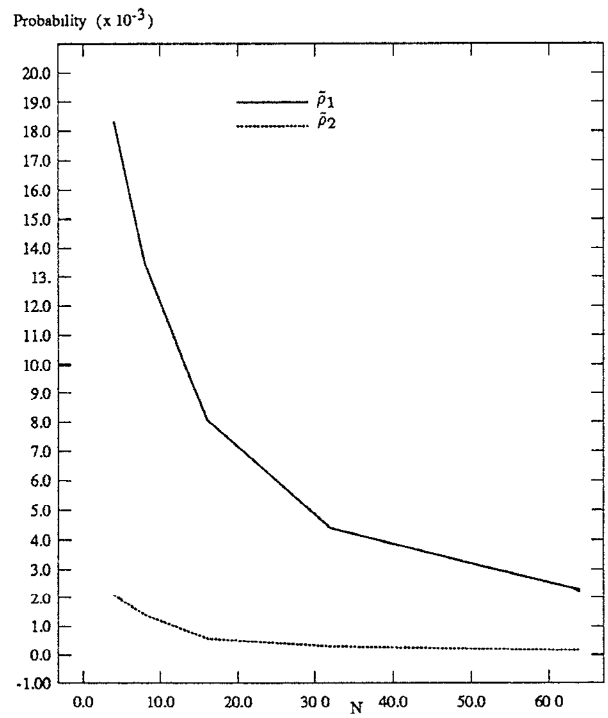

Figure 4: Comparison of two Poisson traffic types $\tilde{\rho}_1$ and $\tilde{\rho}_2$ with bandwidth requirements $a_1 = 1$ (one connection per arrival) and $a_2 = 2$ (two connections per arrival)