UNE IMP

EHE CWI

QMIPS

INR UER

ZAR UTO

Quantitative Modeling in Parallel Systems

An Approximate Analysis of Asynchronous, Packet-switched Buffered Banyan Networks with Blocking

Peter G. Harrison Afonso de C. Pinto
Department of Computing
Imperial College
180 Queen's Gate, London SW7 2BZ

D W2.T1.10

22 September 1994

Distribution Level: All partners

Approved by: Peter G. Harrison

Abstract An approximate algorithm is developed for the performance analysis of buffered, packet-switched, asynchronous networks with no feedback. We focus on banyan networks which are important in parallel computer architectures and telecommunication systems to illustrate our approach and our algorithm is tailored to this problem. The networks we consider are organised in a finite number of stages through each of which a task passes successively in its transmission. The components which forward messages in each stage are modelled independently as small sub-networks of queues; these components are crossbars in the case of banyan networks. Blocking can occur on several upstream paths preceding a full component and the inter-stage blocking effect is taken into account iteratively, blocking probabilities being computed in terms of results of a previous step. The networks we consider may be non-homogeneous in that different servers may have different rates and the arrival processes need not be identical. We derive the queue length probability distribution of each switch from which performance measures, such as blocking probability, mean buffer occupancy and mean transmission time, follow.

The QMIPS Project The QMIPS project is a collaborative research project supported by the CEC as ESPRIT-BRA project no 7269. It is being carried out by the following organisations: CWI (Amsterdam), EHEI (University of Paris V), Imperial College (London), INRIA (Sophia-Antipolis), University of Erlangen, University of Newcastle, University of Torino and University of Zaragoza.

1 Introduction

As computer and communication technology continues to advance, there is an increasing need for efficient communication networks. As traffic increases in asynchronous, packet-switched networks, blocking occurs at the nodes and it is this problem that we address. Our analysis applies to networks with no feedback which can always be structured in stages, some of which can be skipped by a packet. The components which forward messages in each stage are modelled independently as small sub-networks of queues; these components are *crossbar switches* in the case of banyan networks. The approach uses an iterative algorithm of the type used by Jun and Perros [7] for tandem networks but we generalise to feed-forward networks which need not be tandem. If we consider a particular packet, its path through the network can be viewed as an open tandem network with external arrivals at each node. Essentially, it is the handling of these arrivals, which are not independent of the said packet nor of each other, which constitutes the generalisation. The present paper developed from the method described in [4] but uses generalised exponential (GE) random variables to approximate general time delays rather than 2–phase Coxians. This facilitates a better representation of burstiness (GE interarrival times model Poisson bulk arrivals of geometric size) and GE distributions are defined completely by just two moments. However, since zero service times are possible, state transition matrices become much more complex.

Although our analysis applies to any feed-forward network with finite number of stages, we focus on regular banyan networks [2] which form a large class of multi-stage interconnection networks (MINs) and many other MINs can be thought of as being extensions of banyan networks. These networks are especially popular in parallel computer architectures and in the ATM switches prevalent in telecommunication network designs. At each stage of a banyan, there is a set of inputs and a set of outputs. These are numbered (arbitrarily) from the top of a network. The same applies to the switching components in a stage, i.e. the crossbars, and the number of each component is called its level in that stage. We consider regular banyans constructed from 2×2 crossbars. There are a number of different ways of connecting the outputs of one stage of switches to the inputs of the next so as to obtain full connectivity. The corresponding permutation functions define the network topology. All these topologies have been shown to be equivalent by Wu and Feng [26]. Each topology can be obtained from any other by permuting the switches in each stage of the network and so all have the same performance characteristics. Our approach applies not only to networks with a unique path from each input pin to each output, but also to any feedforward network.

Various design issues must be considered when selecting the architecture of a MIN (or any interconnection network, for that matter). Such issues can be identified as: operation mode (synchronous and asynchronous), control strategy (centralised or by the individual switching elements), switching method (circuit switching, packet switching, or a combination of both) and network topology. A considerable amount of research has been reported on specific combinations of the operational characteristics of MINs (e.g., [1, 3, 4, 5, 8, 14, 18, 26]). Patel [19] extensively analysed the case of circuit-switched interconnection networks. The performance of banyan networks with infinite buffer sizes and a particular hot-spot traffic pattern have been reported in [10, 11]. MINs with finite buffers under a uniform traffic pattern have been also studied (e.g., [21, 27]). The single buffer case under uniform traffic pattern has been reported with different approaches in [1, 6, 22]. Results on clocked (synchronous), finite-buffered, packet-switched MINs are given in [14, 15]. To date, results on asynchronous, finite-buffered, packet-switched MINs have not been reported in the literature, to the authors' knowledge. The aim of the present work is to provide an analytical model that can be used to obtain performance measures for asynchronous, finite-buffered, packet-switched MINs with arbitrary traffic pattern, buffer sizes, input rates, and stage-to-stage transmission times.

The analysis and the resulting algorithm we present is based on the following principles. The first step computes the joint probability distribution of the queue-lengths in each component of each stage–j. The iteration begins at the rightmost (i.e., output) stage which is assumed never to be blocked and propagates back to the first. The second step works from left to right and determines the output process of each component in successive stages in terms of its input process using approximate results on the superposition and splitting of arrival processes. The input process to the first stage is given — here Poisson. The mapping function from input to output process at each component in a stage depends on the joint distribution of queue lengths determined in the first step. The two steps are then repeated until convergence is achieved, measured by the closeness of successive approximations of the joint queue length probability distributions. Numerical results and validation of our model with respect to simulation are given in Section 4.

2 Model and Notation

We consider a regular buffered banyan network consisting of M stages and L levels in a packet-switching environment. The basic component is the two-input, two-output switching element (2×2 crossbar switch). For an $N \times N$ network (i.e., N inputs and N outputs), with N a power of 2, it follows that $M = \log_2 N$ and $L = N/2$. An 8×8 banyan network is illustrated in Figure 1. We shall call switch–(i, j) the switching element located at the ith level of the jth stage of the network. The upper input pin of each switch is labeled "input–0" and the lower input pin, "input–1", and similarly for output pins. The topology of the network is given by the matrix $\mathcal{T} = \{t_{ij}\}_{L\times(M-1)}$. The element $t_{ij} = [(a_{ij}, x_{a_{ij}}), (b_{ij}, x_{b_{ij}})]$ is associated with switch–$(i, j+1)$ as follows: the first ordered pair gives the address of the output pin located at the jth stage connected to input–0 of switch–$(i, j+1)$ and the second pair gives the address of the output pin, also located at the jth stage, connected to input–1 of switch–$(i, j+1)$. The component a_{ij} denotes a switch number and $x_{a_{ij}}$ $(= 0, 1)$ denotes an output pin number at that switch. In

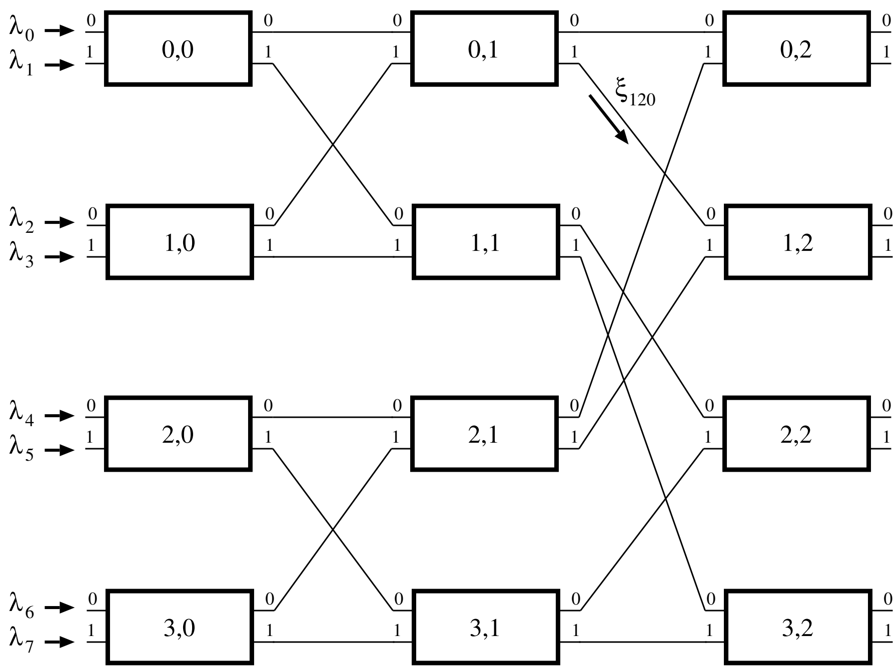

Figure 1: 8×8 Banyan Network.

other words,

$$\begin{array}{lcl} \text{input--0 of switch--}(i, j{+}1) & - & \text{linked to output-}x_{a_{ij}} \text{ of switch--}(a_{ij}, j) \\ \text{input--1 of switch--}(i, j{+}1) & - & \text{linked to output-}x_{b_{ij}} \text{ of switch--}(b_{ij}, j). \end{array}$$

As an example, consider the network shown in Figure 1. In this case, the topology matrix $\boldsymbol{\mathcal{T}}$ is given by:

$$\boldsymbol{\mathcal{T}} = \begin{bmatrix} [(0,0),(1,0)] & [(0,0),(2,0)] \\ [(0,1),(1,1)] & [(0,1),(2,1)] \\ [(2,0),(3,0)] & [(1,0),(3,0)] \\ [(2,1),(3,1)] & [(1,1),(3,1)] \end{bmatrix}$$

The topology of the network can also be given in terms of $\boldsymbol{\mathcal{T}}_b$, the "backward" topology matrix. As opposed to the case of $\boldsymbol{\mathcal{T}}$, its (i,j)th element gives the addresses of input pins in stage–$(j+1)$ connected to the output pins of switch–(i,j). Matrix $\boldsymbol{\mathcal{T}}_b$ can be derived directly from $\boldsymbol{\mathcal{T}}$.

Let $\mathbf{R} = \{r_{ij}\}_{N\times N}$ be the routing probability matrix where r_{ij} is the probability that a task originating at input pin i (first stage) has output pin j (last stage) as its destination, where i, j are numbered from 0 at the top as above. Tasks are assumed to arrive at the switches located in the first stage as a Poisson process with arrival rate λ_i at input pin i. Let then, $\boldsymbol{\lambda} = [\lambda_0, \lambda_1, \ldots, \lambda_{N-1}]$ be the external arrival rate vector. It is assumed that each switch has a single finite buffer with the exception of the first stage switches which are assumed to have a single buffer with infinite size. Let $\boldsymbol{\mathcal{C}}$ be the switch-capacity matrix where the (i,j)th element is given by $\mathcal{C}_{ij}$, the capacity, or buffer size of switch–(i,j), $0 \le i \le L-1$, $0 \le j \le M-1$. Throughout the paper, we shall use $H^*(s)$ to denote the Laplace-Stieltjes transform (LST) of the probability distribution function $H(t)$ of a random variable H, i.e., $H(t) = \mathbf{P}\{H \le t\}$.

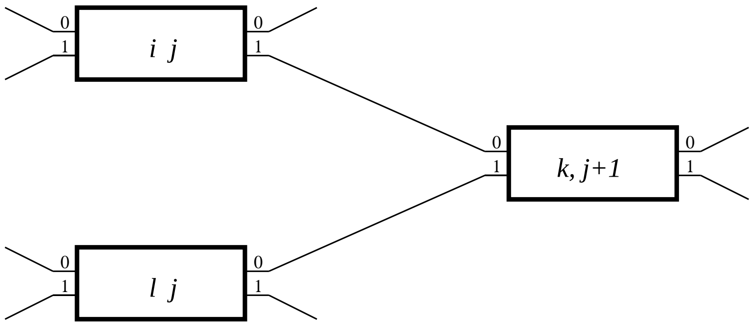

Figure 2: Blocking mechanism.

Incoming tasks join a switch's single buffer and each corresponding output picks the next task requesting its service. In our model each output pin can have any service time distribution. However, for analytical and computational tractability, we use generalised exponential (GE) distributions to approximate general distributions with known first two moments. This type of approximation has been found to be a robust and versatile tool for the analysis of general queueing systems, particularly when bursty traffic is present, [9]. Let $S_{ijx}(t)$ be the service time distribution at output–x of switch–(i,j), $x = 0,1$, $0 \le i \le L-1$, $0 \le j \le M-1$. The distribution $S_{ijx}(t)$ is defined in terms of the parameters μ_{ijx} and θ_{ijx} as follows:

$$S_{ijx}(t) = 1 - \theta_{ijx} e^{-\mu_{ijx} t},$$

with LST $S^*_{ijx}(s)$ given by:

$$\begin{aligned} S^*_{ijx}(s) &= (1-\theta_{ijx}) + \theta_{ijx}\frac{\mu_{ijx}}{s+\mu_{ijx}}, \\ &\qquad 0 \le i \le L-1,\ 0 \le j \le M-1,\ x = 0,1. \end{aligned} \tag{1}$$

We define $\mathbf{S}$ as the 'service-time matrix' whose (i,j)th element, s_{ij}, is given by the two ordered-pairs $[(\mu_{ij0},\theta_{ij0}),(\mu_{ij1},\theta_{ij1})]$ which give the parameters of the service time distributions of outputs 0 and 1 of switch–(i,j), respectively. The switch protocol considered in this paper is presented in detail in Section 2.1.

Every task that arrives at one of the inputs of the first stage, goes to its output destination (last stage) passing through one switch at every stage of the network in a tandem fashion. Consider a task that is being served at, say, output–1 of switch–(i,j) (Figure 2). Suppose that the next switch in this task's path is switch–$(k,j+1)$, without loss of generality, at input–0. When this task completes its service it moves to input–0 of switch–$(k,j+1)$ if there is space available. On the other hand, in the case switch–$(k,j+1)$ is full when the said task completes its service, it has to wait at output–1 of switch–(i,j) until a position in switch–$(k,j+1)$ becomes available. In this situation, output–1 of switch–(i,j) becomes blocked which implies that no service can take place on that output, i.e., the associated server becomes idle. We therefore have *Blocking After Service* (BAS). Now assume that another task located at, say, output–0 of switch–(l,j) completes service, and this output also has switch–$(k,j+1)$ in its path (in this case, input–1). Then, output–0 of switch–(l,j) also becomes blocked and the corresponding server becomes idle, with the result that other tasks that might request that output cannot be served. As soon as a position in switch–$(k,j+1)$ becomes available, the task located at output–1 of switch–(i,j) joins switch–$(k,j+1)$ since it became blocked first. The task located at output–0 of switch–(l,j) only joins switch–$(k,j+1)$ when a second position becomes available. We say that output–1 of switch–(i,j) and output–0 of switch–(l,j) are *neighbours* in the sense that they are both connected to the same switch, but obviously each connected to a different input pin.

The algorithm presented here produces as result, the steady-state joint queue-length probability distribution of each switch of the network. When considering a single switch, both arrival and service processes have to be revised in order to take into account the blocking effect described above and the dependence between the internal arrival processes at the inputs of each switch. Also, the capacity of each switch has to be revised, i.e., increased by two units. This is due to the fact that each switch can block the two output pins that are connected to it as explained above, and the queue-length distribution obtained when analysing a single switch has to reflect this condition.

Since the approximation algorithm presented in Section 3 is based on the analysis of each switch individually, we shall detail the operation of a particular switch.

2.1 Modelling a Single Switch

Tasks arriving at input–0 of a single switch can have either output–0 or output–1 as destination. The same applies to tasks arriving at input–1. In both cases, given that the buffer is not full upon arrival, an incoming task joins the common queue of capacity $\mathcal{C}$ (Figure 3a). Looking at the queue only in terms of output requirements, there are two classes of tasks namely, those that are to be transmitted through output–0 and those to be transmitted through output–1. We say that a service is taking place at output–x, $x = 0, 1$, when a task is being transmitted through that output. Tasks are served on an FCFS basis within their particular classes. Equivalently, we can look at the system as having two queues of size $\mathcal{C}$ each, but such that the total number of tasks present in the switch at any particular time must not exceed $\mathcal{C}$ (Figure 3b). We shall, however, consider the alternative view shown in Figure 3c, which contains characteristics of the previous two. As in (3b), there are two queues with size $\mathcal{C}$ with the restriction that the total number of tasks in the switch be less than or equal to $\mathcal{C}$, and as in (3a), the incoming traffic originating from both input–0 and input–1 merges at point A (Figure 3a) where it then divides according to splitting probabilities which we assume to be independent (Markov routing). The splitting probabilities, u and v, with $u + v = 1$ (Figure 3c), are obtained from the routing probability matrix $\mathbf{R}$, the topology matrix $\boldsymbol{\mathcal{T}}$ and the position of the switch in the network. In Section 3

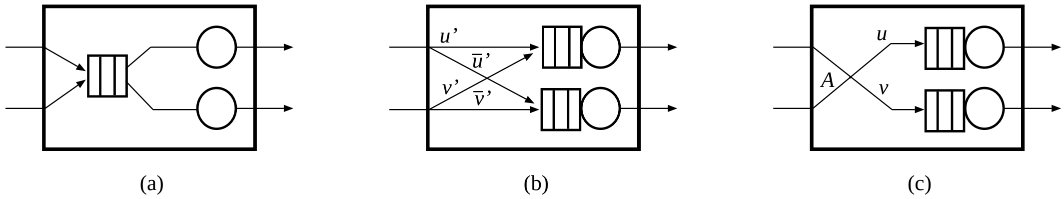

Figure 3: 2×2 Crossbar Switch.

we shall consider splitting and merging in more detail. Here, for the purpose of obtaining the switch's queue-length distribution, we assume that u and v are known, where:

u = probability that an arriving task has to be transmitted through output–0,
v = probability that an arriving task has to be transmitted through output–1.

In the approximation procedure described in Section 3, tasks arrive at each switch according to two different arrival processes depending on the step of the procedure. In the first step (Step–0), which is an initialisation step, the arrival processes at the inputs of all switches of the network are Poisson, whereas in the other steps the interarrival times at the inputs of all switches of the network (except stage–0) are generally distributed, approximated by a GE distribution. Since the buffer capacity of every first-stage switch is assumed to be infinite, every task arrival is accepted into the network. As a consequence, the effective arrival rate at each input pin of any stage can be calculated from the routing probability matrix $\mathbf{R}$, the topology matrix $\boldsymbol{\mathcal{T}}$, and the external arrival rates, $\boldsymbol{\lambda}$. In the case that stage–0 switches have finite capacity, the effective arrival rate at each input pin of these switches cannot be obtained *a priori* and neither can the effective arrival rate at any other input pin of the network. However, an iterative procedure based on the algorithm presented in this paper can be developed for this purpose. Returning to the case of stage–0 switches with infinite capacity, we shall call ξ_{ijx}, the effective arrival rate at input–x of switch–(i,j). As an example, the effective arrival rate ξ_{120} at input–0 of switch–(1,2) in the network shown in Figure 1, is given by $\xi_{120} = \lambda_0(r_{02}+r_{03}) + \lambda_1(r_{12}+r_{13}) + \lambda_2(r_{22}+r_{23}) + \lambda_3(r_{32}+r_{33})$. We assume that the system is stable.

We now describe the behaviour of a switch analysed in isolation for each interarrival distribution considered. We adopt the following notation:

$\mathcal{S}_{2\times2}(A/GE/1/\mathcal{C})$: indicates that interarrival times at a 2×2 switch are exponentially distributed (respectively, GE) when $A = M$ (respectively, $A = GE$), the service time at each one of the output pins is GE distributed, with these distributions not necessarily equal. It also indicates that each pin has only one server and that the switch has total capacity $\mathcal{C}$.

Note that in order to analyse a switch according to either one of these two models, we consider the combined, or superposed arrivals of both input–0 and input–1.

Single Switch - Interarrival Times Exponentially Distributed We now outline the method of obtaining the steady-state joint queue-length distribution of the system $\mathcal{S}_{2\times2}(M/GE/1/\mathcal{C})$ as depicted in Figure 4. The interarrival times of the total arrival process are exponentially distributed with parameter λ at switch population less than $\mathcal{C}-1$ (no blocking), λ' at population $\mathcal{C}-1$ (one input blocked) and 0 at population $\mathcal{C}$ (both inputs blocked). For switch–(i,j), η_{ij} denotes the ratio of these rates, λ'/λ and we define $\eta_{ij} = 2\xi_{ij0}\xi_{ij1}/(\xi_{ij0}+\xi_{ij1})^2$,

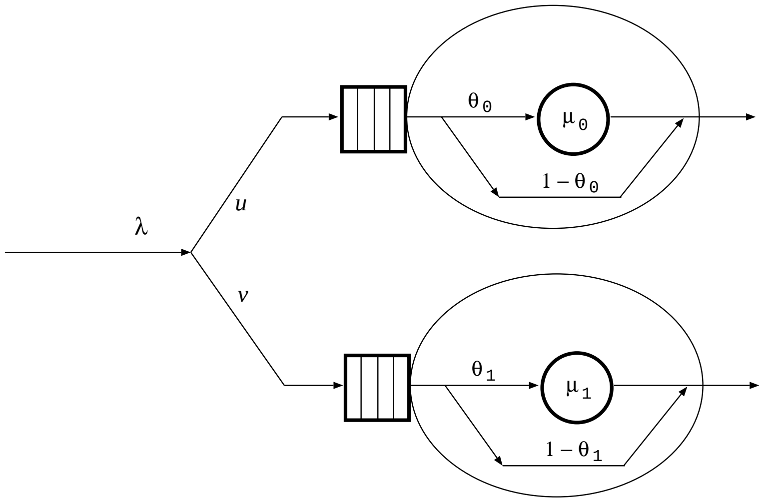

Figure 4: $\mathcal{S}_{2\times 2}(M/GE/1/\mathcal{C})$.

assuming that when exactly one input is blocked, it is input–0 with probability $\xi_{ij0}/(\xi_{ij0}+\xi_{ij1})$ and similarly for input–1.

Actually, the arrival process is a superposition of two Poisson processes originating at input–0 and input–1. It follows that the resulting process is also Poisson, since the superposition of two or more Poisson processes is itself Poisson (see, for instance, [16, pages 45–48]) with rate given by the sum of the rates of the superposed processes. When the total number of tasks in the system is $\mathcal{C}$ (i.e., the system is full), the arrival rate is zero. Service times are assumed to be GE distributed. We shall call queue–x the queue of tasks that are served at output–x, $x = 0, 1$. We define the following parameters:

μ_x	:	service rate of queue–x, $x = 0, 1$,
θ_x	:	probability that an arriving task, immediately after entering the service centre, remains for an exponential service time, $x = 0, 1$,
$1-\theta_x$	:	probability that an arriving task, immediately after entering the service centre, leaves without being served, $x = 0, 1$.

Let the random variables $N_x(t)$, $x = 0, 1$, be the number of tasks located at queue–x (including the one being served) at time t. The state of the system at time t is given by the vector $(N_0(t), N_1(t))$ and the state space $\Omega_0 = \{(n_0, n_1) |\ 0 \le i \le \mathcal{C},\ 0 \le j \le \mathcal{C} - i\}$. The steady-state queue-length distribution of $\mathcal{S}_{2\times 2}(M/GE/1/\mathcal{C})$ is denoted by $p(n_0, n_1)$ for $(n_0, n_1) \in \Omega_0$, i.e.,

$$p(n_0, n_1) = \lim_{t\to\infty} \mathbf{P}\{N_0(t) = n_0, N_1(t) = n_1\}.$$

Since we have a finite state space continuous time Markov chain, there exists a steady-state solution for the $p(n_0, n_1)$, obtained by solving the system's balance equations. Consider one of the balance equations of the system $\mathcal{S}_{2\times 2}(M/GE/1/\mathcal{C})$ given below:

$$p(0,0)[\lambda v\theta_1 + \lambda u\theta_0] = \mu_0 \sum_{i=1}^{\mathcal{C}} p(i,0)(1-\theta_0)^{i-1} + \mu_1 \sum_{j=1}^{\mathcal{C}} p(0,j)(1-\theta_1)^{j-1}.$$

The left-hand side of the equation represents the flow out of state $(0,0)$ in steady-state. A transition can occur due to an external arrival which either decides to go to queue–0 with probability u and start service (with probability θ_0), or decides to go to queue–1 with probability

v and start service (with probability θ_1). The right-hand side represents the flow into state $(0,0)$ in steady-state. The change of state in this case occurs due to one service completion at either queue and the subsequent decision of the remaining tasks located at that queue to leave the system without being served (with probability $(1-\theta_x)$, $x = 0,1$) giving a bulk departure. The complete system of balance equations for $\mathcal{S}_{2\times2}(M/GE/1/\mathcal{C})$ is given in Appendix A. Its solution is obtained numerically.

Single Switch - GE Interarrival Times The $\mathcal{S}_{2\times2}(GE/GE/1/\mathcal{C})$ queueing system is shown in Figure 5. This system is practically the same as the one analysed above apart from

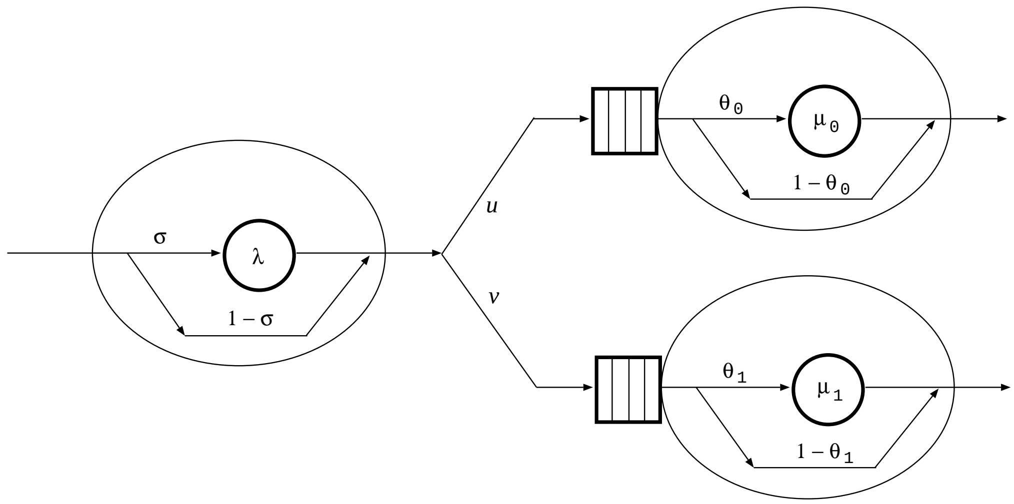

Figure 5: $\mathcal{S}_{2\times2}(GE/GE/1/\mathcal{C})$.

the fact that the interarrival times are, in this case, GE distributed with parameters (λ, σ) at switch population less than $\mathcal{C}-1$, (λ', σ) at population $\mathcal{C}-1$ and 0 at $\mathcal{C}$. We again assume, as for the case of Poisson arrivals, that $\lambda' = \eta_{ij}\lambda$ for switch–(i,j).

Again the arrival process is a superposition of two arrival processes originating at input–0 and input–1. In this case, they both have GE interarrival times. We make use of an approximation technique (Section 3.2.2) to obtain a GE distribution for the superposed arrival process's interarrival times. The state of the $\mathcal{S}_{2\times2}(GE/GE/1/\mathcal{C})$ system at time t is again given by the vector $(N_0(t), N_1(t))$ and the steady-state queue-length distribution is denoted by $p(n_0, n_1)$ where $(n_0, n_1) \in \Omega_0$. Here we also obtain $p(n_0, n_1)$ by solving the system's balance equations. Consider one of the balance equations of the system $\mathcal{S}_{2\times2}(GE/GE/1/\mathcal{C})$ given below:

$$p(0,0)\lambda[1 - \frac{u\sigma(1-\theta_0)}{\theta_0 + \sigma - \sigma\theta_0} - \frac{v\sigma(1-\theta_1)}{\theta_1 + \sigma - \sigma\theta_1}] =$$
$$\mu_0 \sum_{k=1}^{\mathcal{C}} p(k,0)(1-\theta_0)^{k-1} + \mu_1 \sum_{k=1}^{\mathcal{C}} p(0,k)(1-\theta_1)^{k-1}.$$

The left-hand side of the equation represents the flow out of state $(0,0)$ in steady-state. A transition can occur due to an external bulk arrival with size geometrically distributed with parameter σ. We assume that all tasks in an arriving batch go to queue–0 with probability u, or go to queue–1 with probability v, i.e., tasks in the same batch do not go to different queues. The expression for the flow out of state $(0,0)$ is obtained as follows:

$$\text{"flow out of state }(0,0)\text{"} = \lambda\sigma \sum_{k=1}^{\infty} (1-\sigma)^{k-1} [u(1-(1-\theta_0)^k) + v(1-(1-\theta_1)^k)],$$

where $\sigma(1-\sigma)^{k-1}$ is the probability that an arriving batch has size k, $u(1-(1-\theta_0)^k)$ is the probability that the batch goes to output–0 and at least one task receives service, and $v(1-(1-\theta_1)^k)$ is the probability that the batch goes to output–1 and at least one task receives service. After some algebraic manipulation this expression reduces to the left-hand side of the balance equation given above. The right-hand side represents the flow into state $(0,0)$ in steady-state and is the same as for the $\mathcal{S}_{2\times2}(M/GE/1/\mathcal{C})$ system. The change of state in this case occurs due to one service completion at either queue and the subsequent decision of the remaining tasks located at that queue to leave the system without being served, with probability $(1-\theta_x)$, $x=0,1$ (bulk departure).

In other balance equations, transitions of the form $(0,k) \to (j,k)$ occur (and similarly $(k,0) \to (k,j)$). These arise from a number b of tasks arriving at output–0 and skipping service, followed by a task which enters service at output–0 and $j-1$ further tasks which queue. Thus the corresponding transition rate is

$$\lambda u\sigma\theta_0 \sum_{b=0}^{\infty}(1-\sigma)^{b+j-1}(1-\theta_0)^b = \frac{\lambda u\sigma\theta_0(1-\sigma)^{j-1}}{\theta_0+\sigma-\sigma\theta_0}.$$

The complete system of balance equations for $\mathcal{S}_{2\times2}(GE/GE/1/\mathcal{C})$ is given in Appendix B. As in the case of the $\mathcal{S}_{2\times2}(M/GE/1/\mathcal{C})$ system, the solution of these balance equations is obtained numerically.

3 The Approximation Procedure

In this section we describe the approximation algorithm for obtaining the steady-state joint queue-length distribution of each 'building block' (switch) of a regular banyan network. Before we proceed, we introduce the following notation used in the approximation procedure (the tasks in a queue include any task in service).

$p_{ij}^{(k)}(n_0,n_1)$	:	approximation obtained in Step–k, of the stationary probability that there are n_0 tasks at queue–0 and n_1 tasks at queue–1 of switch–(i,j), $k=0,1,2$,
$p_{ij}^{(k)}(n)$	:	approximation obtained in Step–k, of the stationary probability that there are n tasks in switch–(i,j), $k=0,1,2$.

3.1 Step–0

This is an initialisation step which gives a first approximation for the joint queue-length distribution at the queues located in each switch of the network. The main result obtained in Step–0 is the distribution $p_{ij}^{(0)}(n_0,n_1)$, $0 \le n_0 \le \mathcal{C}_{ij}+2$, $0 \le n_1 \le \mathcal{C}_{ij}+2-n_0$. Note that two units are added to the capacity of the switch–(i,j) to reflect the two tasks of the previous stage that can be blocked by switch–(i,j). Initially all the characteristics of the network, $(\boldsymbol{\lambda}, \boldsymbol{\mathcal{C}}, \boldsymbol{\mathcal{T}}, \mathbf{R}, \mathbf{S})$, are read in the form of input data. Then, the effective arrival rate at every input pin of the network, ξ_{ijx} is obtained, $0 \le i \le L-1$, $0 < j \le M-1$, $x=0,1$. The approach used in this step of the algorithm is that initially, all switches of the last stage (stage–$(M-1)$) are analysed as $\mathcal{S}_{2\times2}(M/GE/1/\mathcal{C}_{i,M-1}+2)$ systems and then the algorithm proceeds backwards to switches of stage–0. At each stage of the network, the corresponding switches are all treated as $\mathcal{S}_{2\times2}(M/GE/1/\mathcal{C}_{ij}+2)$ but unlike the switches of stage–$(M-1)$, the distribution of service time and arrival rate at each queue have to be revised in order to reflect the effect of blocking. Let $B_{ijx}^*(s)$ be the LST of $B_{ijx}(t)$, the *effective* service time distribution at output–x of switch–(i,j), $0 \le i \le L-1$, $0 \le j \le M-1$, $x=0,1$. From $B_{ijx}^*(s)$ we obtain the first two moments of the effective service time distribution, which can then be approximated by a GE distribution, with parameters $\mathbf{br}_{ijx} = (b_{ijx}, \beta_{ijx})$ say. Note that each stage–0 switch is analysed as a pair of independent $M/GE/1/\infty$ queues, since we are assuming that all stage–0 switches have infinite capacity.

Since all switches located at the last stage of the network cannot be blocked, they can be analysed as $\mathcal{S}_{2\times2}(M/GE/1/\mathcal{C}_{i,M-1}+2)$ systems with arrival rate $\lambda_{i,M-1} = \xi_{i,M-1} = \xi_{i,M-1,0}+\xi_{i,M-1,1}$ (and $\lambda'_{i,M-1} = \eta_{i,M-1}\lambda_{i,M-1}$), and effective service time distributions at queue–0 and queue–1, given by the LST's $B^*_{i,M-1,0}(s) = S^*_{i,M-1,0}(s)$ and $B^*_{i,M-1,1}(s) = S^*_{i,M-1,1}(s)$, respectively, $0 \le i \le L-1$. Note that no stage–$(M-1)$ output has a *neighbour* output in the sense described in Section 2.

Now let's consider switch–$(k, j-1)$ located in an intermediate stage. Suppose one of the outputs of switch–$(k, j-1)$ is connected to one of the inputs of switch–(i, j), i.e., let input–0 and input–1 of switch–(i, j) be connected to output–x of switch–$(k, j-1)$ and output–y of switch–$(l, j-1)$, respectively, with x and y being either 0 or 1, (not necessarily different), for some k, l, $0 \le k, l \le L-1$ (see Figure 6). It follows that these two outputs are *neighbours*, i.e., they

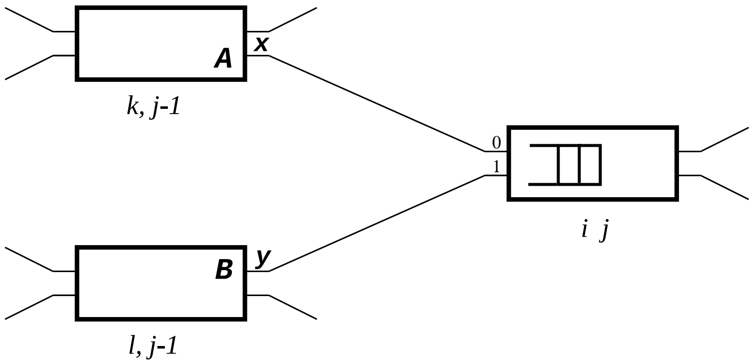

Figure 6: Effective Service Time.

are both connected to input pins of the same switch. We label a task located at queue–x of switch–$(k, j-1)$ *task* A and a task located at output–y of switch–$(l, j-1)$ *task* B. Since tasks A and B are located at *neighbour* outputs, we say that they are *neighbour* tasks.

Now, upon its service completion, task A can find its destination switch (switch–(i, j)) in three different states in terms of the total number of tasks there: (i) less than its capacity, $\mathcal{C}_{ij}$, (ii) equal to $\mathcal{C}_{ij}$, and (iii) equal to $\mathcal{C}_{ij} + 1$. In the first case, task A simply occupies the next available position of switch–(i, j)'s buffer. The second case indicates that task A finds switch–(i, j) completely full and that task A, being the first to be blocked, will consequently be the first to enter switch–(i, j)'s buffer whenever a position becomes available. During this interval, task A occupies the fictitious position $\mathcal{C}_{ij} + 1$. Finally, the third case indicates that task A finds a task B already blocked by switch–(i, j), which implies that task A has to await two service completions of tasks located at switch–(i, j) in order to enter its destination buffer. Task A occupies the fictitious position $\mathcal{C}_{ij} + 2$ during this time interval. It follows that the revised service time distribution of task A must reflect the additional delay, if any, corresponding to each one of these three situations. Note that it is not possible for task A to find $\mathcal{C}_{ij} + 2$ tasks in switch–(i, j) when it finishes its service. This is due to the fact that when the fictitious position $\mathcal{C}_{ij} + 2$ is occupied, all tasks located at both outputs connected to switch–(i, j) are blocked, which means that no service can take place at either one of these outputs.

Let $B^*_{k,j-1,x}(s|\cdot)$ denote the LST of the distribution of the effective service time at queue–x of switch–$(k, j-1)$ *conditioned* on the number of tasks located at its destination switch at the time a task located at queue–x completes its service. The cases (i), (ii) and (iii) mentioned above correspond to $B^*_{k,j-1,x}(s| < \mathcal{C}_{ij})$, $B^*_{k,j-1,x}(s| = \mathcal{C}_{ij})$, and $B^*_{k,j-1,x}(s| = \mathcal{C}_{ij} + 1)$, respectively.

For the case that, upon service completion, task A finds less than $\mathcal{C}_{ij}$ tasks in switch–(i,j), it is clear that there is no additional delay caused by the state of switch–(i,j), which implies that the effective service time distribution of task A must be equal to its original service time distribution, i.e.,

$$B^*_{k,j-1,x}(s| < \mathcal{C}_{ij}) = S^*_{k,j-1,x}(s), \qquad 0 \le i \le L-1,\ \ 0 < j < M-1. \tag{2}$$

If however, at the time task A completes its service, the number of tasks located at switch–(i,j) is $\mathcal{C}_{ij}$, then the effective service time of task A is the sum of its original service time and the residual service time in switch–(i,j). Note that, at this point, the effective service time distribution of both queues of switch–(i,j) have been approximated by GE distributions. Now, assume first that switch–(i,j) is in state $(\mathcal{C}_{ij},0)$ at the time task A completes its service. Then the residual service time in switch–(i,j) is the residual service time at queue–0 of switch–(i,j). Since a non-zero service time is exponentially distributed, the residual effective service time is exponentially distributed with parameter b_{ij0}. Assuming now that the state of the switch is $(0,\mathcal{C}_{ij})$, and following the above argument, a corresponding result for queue–1 is obtained. Finally, if the switch is in one of the states $(n, \mathcal{C}_{ij}-n)$, $1 \le n \le \mathcal{C}_{ij}-1$, i.e., both outputs are busy, then the residual service time of switch–(i,j) is the minimum of the residual service times of queue–0 and queue–1. In this case, the residual service time is the minimum of two exponential random variables with parameters b_{ij0} and b_{ij1}, which is also exponential, with parameter $b_{ij0}+b_{ij1}$. We can now obtain $B^*_{k,j-1,x}(s| = \mathcal{C}_{ij})$ as follows:

$$\begin{aligned} B^*_{k,j-1,x}(s| = \mathcal{C}_{ij}) \;&=\; S^*_{k,j-1,x}(s) \cdot \{\gamma^{(1)}_{ij}(\frac{b_{ij0}}{s+b_{ij0}}) + \gamma^{(2)}_{ij}(\frac{b_{ij1}}{s+b_{ij1}}) + \gamma^{(3)}_{ij}(\frac{b_{ij0}+b_{ij1}}{s+b_{ij0}+b_{ij1}})\}, \\ & \qquad\qquad 0 \le i \le L-1,\ \ 0 < j < M-1, \end{aligned} \tag{3}$$

where the probabilities $\gamma^{(1)}_{ij}$, $\gamma^{(2)}_{ij}$, and $\gamma^{(3)}_{ij}$ are defined as follows:

$\gamma^{(1)}_{ij}$	:	conditional probability that there are no tasks in queue–1 of switch–(i,j), given that switch–(i,j) is full (i.e., queue–0 has $\mathcal{C}_{ij}$ tasks).
$\gamma^{(2)}_{ij}$	:	conditional probability that there are no tasks in queue–0 of switch–(i,j), given that switch–(i,j) is full (i.e., queue–1 has $\mathcal{C}_{ij}$ tasks).
$\gamma^{(3)}_{ij}$	:	conditional probability that there is at least one task in each queue of switch–(i,j), given that switch–(i,j) is full.

Although the probabilities $\gamma^{(1)}_{ij}, \gamma^{(2)}_{ij}$, and $\gamma^{(3)}_{ij}$ are conditioned on an arrival instant we can calculate them approximately from the steady-state distribution $p_{ij}(n_0,n_1)$, making use of the PASTA argument ([25]), considering the fact that in this step, the arrival process to each input is assumed to be Poisson. Thus,

$$\gamma^{(1)}_{ij} \;=\; \frac{p_{ij}(\mathcal{C}_{ij},0)}{p_{ij}(\mathcal{C}_{ij})}, \qquad \gamma^{(2)}_{ij} \;=\; \frac{p_{ij}(0,\mathcal{C}_{ij})}{p_{ij}(\mathcal{C}_{ij})}, \qquad \gamma^{(3)}_{ij} \;=\; \frac{\sum_{n=1}^{\mathcal{C}_{ij}-1} p_{ij}(n, \mathcal{C}_{ij}-n)}{p_{ij}(\mathcal{C}_{ij})},$$

where $p_{ij}(\mathcal{C}_{ij}) = \sum_{n=0}^{\mathcal{C}_{ij}} p_{ij}(n, \mathcal{C}_{ij}-n)$. We omit the superscript indicating the step of the algorithm for notational simplicity. Note that at this point in the algorithm, where we are analysing switches located at the $(j-1)$th stage, all the stages to the right of stage–$(j-1)$ have already been analysed (since Step–0 moves from right to left). Therefore, we can make use of any of the distributions p_{il}, with $l = j, j+1, \ldots, M-1$.

Finally, upon service completion of task A, if the number of tasks located at switch–(i,j) is $\mathcal{C}_{ij}+1$ (which implies that there is a task B already blocked), then the effective service time of task A is the sum of its original service time, the residual service time in switch–(i,j) and an additional service time in switch–(i,j) which starts at the time task B leaves the fictitious position $\mathcal{C}_{ij}+1$ and joins switch–(i,j). Before we proceed, we define the following:

- $\mathcal{G}_{ij}^{(0)*}(s)$: LST of the distribution of the minimum of two exponential r.v's with parameters b_{ij0} and b_{ij1}.
- $\mathcal{G}_{ij}^{(1)*}(s)$: LST of the distribution of the minimum of a GE r.v. with parameters (b_{ij0}, β_{ij0}) and an exponential r.v. with parameter b_{ij1}.
- $\mathcal{G}_{ij}^{(2)*}(s)$: LST of the distribution of the minimum of an exponential r.v. with parameter b_{ij0} and a GE r.v. with parameters (b_{ij1}, β_{ij1}).
- $\mathcal{G}_{ij}^{(3)*}(s)$: LST of the distribution of the minimum of two GE r.v's with parameters (b_{ij0}, β_{ij0}) and (b_{ij1}, β_{ij1}).

$\mathcal{G}_{ij}^{(k)*}(s)$, $k = 0,1,2,3$, is therefore given by:

$$\begin{aligned}
\mathcal{G}_{ij}^{(0)*}(s) &= \frac{b_{ij0}+b_{ij1}}{s+b_{ij0}+b_{ij1}}, \\
\mathcal{G}_{ij}^{(1)*}(s) &= (1-\beta_{ij0}) + \beta_{ij0}(\frac{b_{ij0}+b_{ij1}}{s+b_{ij0}+b_{ij1}}), \\
\mathcal{G}_{ij}^{(2)*}(s) &= (1-\beta_{ij1}) + \beta_{ij1}(\frac{b_{ij0}+b_{ij1}}{s+b_{ij0}+b_{ij1}}), \\
\mathcal{G}_{ij}^{(3)*}(s) &= (1-\beta_{ij0}\beta_{ij1}) + \beta_{ij0}\beta_{ij1}(\frac{b_{ij0}+b_{ij1}}{s+b_{ij0}+b_{ij1}}). \qquad (4)
\end{aligned}$$

There are eight possible combinations of residual service time and additional service time depending on the state of switch–(i,j) seen by task A at its service completion instant and at the time it leaves fictitious position $\mathcal{C}_{ij}+2$ (i.e., end of service in one of the outputs of switch–(i,j)), respectively. The combinations are:

1. All $\mathcal{C}_{ij}+1$ tasks in switch–(i,j) upon task A's arrival are to be served at output–0 of switch–(i,j). Then both residual service time and additional service time are from output–0.

2. All $\mathcal{C}_{ij}+1$ tasks in switch–(i,j) upon task A's arrival are to be served at output–1 of switch–(i,j). Then both residual service time and additional service time are from output–1.

3. Task A sees switch–(i,j) in state $(\mathcal{C}_{ij},1)$ upon arrival *and* the only task located at queue–1 is the one already blocked. Then, the residual service time is obtained from an output–0 service time, and the additional service time is given by $\mathcal{G}_{ij}^{(3)*}(s)$, because two new service periods begin after the residual service time.

4. Task A sees switch–(i,j) in state $(1,\mathcal{C}_{ij})$ upon arrival *and* the only task located at queue–0 is the one already blocked. Then, the residual service time is obtained from an output–1 service time, and, as in 3, the additional service time is given by $\mathcal{G}_{ij}^{(3)*}(s)$.

5. Task A sees switch–(i,j) in state $(\mathcal{C}_{ij},1)$ upon arrival *and* the task already blocked is to be served by output–0 *and* the first task to complete its service is the one located at output–1. Then the residual service time is given by $\mathcal{G}_{ij}^{(0)*}(s)$, and the additional service time is a residual service time at output–0, i.e., exponential with parameter b_{ij0}.

6. Task A sees switch–(i,j) in state $(1,\mathcal{C}_{ij})$ upon arrival *and* the task already blocked is to be served by output–1 *and* the first task to complete its service is the one located at output–0. Then the residual service time is given by $\mathcal{G}_{ij}^{(0)*}(s)$, and the additional service time is a residual service time at output–1, i.e., exponential with parameter b_{ij1}.

7. Task A sees switch–(i,j) in state $(\mathcal{C}_{ij},1)$ upon arrival *and* the task already blocked is to be served by output–0 *and* the first task to complete its service is at output–0, *OR* upon arrival, task A sees switch–(i,j) in one of the states $(n,\mathcal{C}_{ij}-n+1)$, $2\le n\le \mathcal{C}_{ij}-1$, *and* the first task to complete its service is at output–0. Then the residual service time is given by $\mathcal{G}_{ij}^{(0)*}(s)$, and the additional service time is given by $\mathcal{G}_{ij}^{(1)*}(s)$.
8. Task A sees switch–(i,j) in state $(1,\mathcal{C}_{ij})$ upon arrival *and* the task already blocked is to be served by output–1 *and* the first task to complete its service is at output–1, *OR* upon arrival, task A sees switch–(i,j) in one of the states $(n,\mathcal{C}_{ij}-n+1)$, $2\le n\le \mathcal{C}_{ij}-1$, *and* the first task to complete its service is at output–1. Then the residual service time is given by $\mathcal{G}_{ij}^{(0)*}(s)$, and the additional service time is given by $\mathcal{G}_{ij}^{(2)*}(s)$.

Thus, the LST $B^*_{k,j-1,x}(s|=\mathcal{C}_{ij}+1)$ can be obtained as follows:

$$\begin{aligned}
B^*_{k,j-1,x}(s|=\mathcal{C}_{ij}+1) \quad = \quad & S^*_{k,j-1,x}(s)\cdot\{\Upsilon_{ij}^{(1)}(\frac{b_{ij0}}{s+b_{ij0}})B^*_{ij0}(s)+\Upsilon_{ij}^{(2)}(\frac{b_{ij1}}{s+b_{ij1}})B^*_{ij1}(s)\\
&+\Upsilon_{ij}^{(3)}(\frac{b_{ij0}}{s+b_{ij0}})\mathcal{G}_{ij}^{(3)*}(s)+\Upsilon_{ij}^{(4)}(\frac{b_{ij1}}{s+b_{ij1}})\mathcal{G}_{ij}^{(3)*}(s)\\
&+\Upsilon_{ij}^{(5)}\mathcal{G}_{ij}^{(0)*}(s)(\frac{b_{ij0}}{s+b_{ij0}})+\Upsilon_{ij}^{(6)}\mathcal{G}_{ij}^{(0)*}(s)(\frac{b_{ij1}}{s+b_{ij1}})\\
&+\Upsilon_{ij}^{(7)}\mathcal{G}_{ij}^{(0)*}(s)\mathcal{G}_{ij}^{(1)*}(s)+\Upsilon_{ij}^{(8)}\mathcal{G}_{ij}^{(0)*}(s)\mathcal{G}_{ij}^{(2)*}(s)\},\\
&\qquad 0\le i\le L-1,\;\; 0<j<M-1. \qquad (5)
\end{aligned}$$

The probabilities $\Upsilon_{ij}^{(k)}$, $k=1,\ldots,8$, correspond to the above mentioned conditions, and are obtained approximately from the iteration as follows:

$$\begin{aligned}
\Upsilon_{ij}^{(1)} &= \frac{p_{ij}(\mathcal{C}_{ij}+1,0)}{p_{ij}(\mathcal{C}_{ij}+1)},\\
\Upsilon_{ij}^{(2)} &= \frac{p_{ij}(0,\mathcal{C}_{ij}+1)}{p_{ij}(\mathcal{C}_{ij}+1)},\\
\Upsilon_{ij}^{(3)} &= \frac{p_{ij}(\mathcal{C}_{ij},1)}{p_{ij}(\mathcal{C}_{ij}+1)}\cdot\frac{p_{ij}(\mathcal{C}_{ij},0)l_{ij}}{[p_{ij}(\mathcal{C}_{ij},0)l_{ij}+p_{ij}(\mathcal{C}_{ij}-1,1)u_{ij}]},\\
\Upsilon_{ij}^{(4)} &= \frac{p_{ij}(1,\mathcal{C}_{ij})}{p_{ij}(\mathcal{C}_{ij}+1)}\cdot\frac{p_{ij}(0,\mathcal{C}_{ij})u_{ij}}{[p_{ij}(0,\mathcal{C}_{ij})u_{ij}+p_{ij}(1,\mathcal{C}_{ij}-1)l_{ij}]},\\
\Upsilon_{ij}^{(5)} &= \frac{p_{ij}(\mathcal{C}_{ij},1)}{p_{ij}(\mathcal{C}_{ij}+1)}\cdot\frac{p_{ij}(\mathcal{C}_{ij}-1,1)u_{ij}}{[p_{ij}(\mathcal{C}_{ij},0)l_{ij}+p_{ij}(\mathcal{C}_{ij}-1,1)u_{ij}]}\cdot\frac{b_{ij1}}{(b_{ij0}+b_{ij1})},\\
\Upsilon_{ij}^{(6)} &= \frac{p_{ij}(1,\mathcal{C}_{ij})}{p_{ij}(\mathcal{C}_{ij}+1)}\cdot\frac{p_{ij}(1,\mathcal{C}_{ij}-1)l_{ij}}{[p_{ij}(0,\mathcal{C}_{ij})u_{ij}+p_{ij}(1,\mathcal{C}_{ij}-1)l_{ij}]}\cdot\frac{b_{ij0}}{(b_{ij0}+b_{ij1})},\\
\Upsilon_{ij}^{(7)} &= \frac{p_{ij}(\mathcal{C}_{ij},1)}{p_{ij}(\mathcal{C}_{ij}+1)}\cdot\frac{p_{ij}(\mathcal{C}_{ij}-1,1)u_{ij}}{[p_{ij}(\mathcal{C}_{ij},0)l_{ij}+p_{ij}(\mathcal{C}_{ij}-1,1)u_{ij}]}\cdot\frac{b_{ij0}}{(b_{ij0}+b_{ij1})}\\
&\quad+\sum_{n=2}^{\mathcal{C}_{ij}-1}\frac{p_{ij}(n,\mathcal{C}_{ij}-n+1)}{p_{ij}(\mathcal{C}_{ij}+1)}\cdot\frac{b_{ij0}}{(b_{ij0}+b_{ij1})},\\
\Upsilon_{ij}^{(8)} &= \frac{p_{ij}(1,\mathcal{C}_{ij})}{p_{ij}(\mathcal{C}_{ij}+1)}\cdot\frac{p_{ij}(1,\mathcal{C}_{ij}-1)l_{ij}}{[p_{ij}(0,\mathcal{C}_{ij})u_{ij}+p_{ij}(1,\mathcal{C}_{ij}-1)l_{ij}]}\cdot\frac{b_{ij1}}{(b_{ij0}+b_{ij1})},\\
&\quad+\sum_{n=2}^{\mathcal{C}_{ij}-1}\frac{p_{ij}(n,\mathcal{C}_{ij}-n+1)}{p_{ij}(\mathcal{C}_{ij}+1)}\cdot\frac{b_{ij1}}{(b_{ij0}+b_{ij1})},
\end{aligned}$$

where u_{ij} and l_{ij} are the probabilities that task B (the one blocked before task A) decides to go to the "upper" or "lower" output of switch–(i,j) when joining the fictitious position $\mathcal{C}_{ij}+1$, respectively. These probabilities are obtained from the routing probability matrix, $\mathbf{R}$.

Note that in order to obtain the LST's of the conditional effective service time distributions, $B^*_{k,j-1,x}(s| < \mathcal{C}_{ij})$, $B^*_{k,j-1,x}(s| = \mathcal{C}_{ij})$, and $B^*_{k,j-1,x}(s| = \mathcal{C}_{ij}+1)$, we need the unconditional effective service time distributions of both outputs of switch–(i,j) (see Equations 3 and 5). Considering queue–x of switch–$(k, j-1)$, the unconditional effective service time distribution $B^*_{k,j-1,x}(s)$ can be calculated as follows:

$$\begin{aligned} B^*_{k,j-1,x}(s) \;&=\; (1-\pi_{ij}-\Pi_{ij})\cdot B^*_{k,j-1,x}(s| < \mathcal{C}_{ij}) \\ &\quad + \pi_{ij}\cdot B^*_{k,j-1,x}(s| = \mathcal{C}_{ij}) + \Pi_{ij}\cdot B^*_{k,j-1,x}(s| = \mathcal{C}_{ij}+1) \\ &\qquad\qquad 1\le k\le L,\;\; 2\le j\le M,\;\; x=0,1, \end{aligned} \tag{6}$$

where the probabilities π_{ij} and Π_{ij} are defined as:

π_{ij} : probability that there are $\mathcal{C}_{ij}$ tasks in switch–(i,j) at the time an incoming task completes its service,

Π_{ij} : probability that there are $\mathcal{C}_{ij}+1$ tasks in switch–(i,j) (i.e., one already blocked) at the time an incoming task completes its service.

Considering switch–(i,j), the two fictitious positions $\mathcal{C}_{ij}+1$ and $\mathcal{C}_{ij}+2$ do not belong to either queue–0 or queue–1 but rather, they form a third "virtual" queue of size two. So, if a task is located at position $\mathcal{C}_{ij}+1$, it can be seen as being in the first position of this third queue, and if it is in position $\mathcal{C}_{ij}+2$, in its second or rather, its last position. Applying Little's result on the second position of this "virtual" queue (i.e., position $\mathcal{C}_{ij}+2$), we obtain the following:

$$p_{ij}(\mathcal{C}_{ij}+2) = [(\xi_{ij0}+\xi_{ij1})\cdot\Pi_{ij}]\cdot\tilde{\mathcal{R}}_{ij}, \tag{7}$$

where Π_{ij} is regarded as a "thinning" probability with respect to the arrival process with arrival rate $\xi_{ij0}+\xi_{ij1}$ and $\tilde{\mathcal{R}}_{ij}$ is the average remaining service time of switch–(i,j) at the time one of the outputs connected to switch–(i,j) gets blocked, given that the other output is already blocked. $\tilde{\mathcal{R}}_{ij}$ is given by:

$$\tilde{\mathcal{R}}_{ij} = \frac{(\Upsilon^{(1)}_{ij}+\Upsilon^{(3)}_{ij})}{b_{ij0}} + \frac{(\Upsilon^{(2)}_{ij}+\Upsilon^{(4)}_{ij})}{b_{ij1}} + \frac{(\Upsilon^{(5)}_{ij}+\Upsilon^{(6)}_{ij}+\Upsilon^{(7)}_{ij}+\Upsilon^{(8)}_{ij})}{b_{ij0}+b_{ij1}}. \tag{8}$$

Once $\tilde{\mathcal{R}}_{ij}$ is calculated, the probability Π_{ij} can be obtained from Equation 7.

In order to calculate the probability π_{ij}, we apply Little's result on the "virtual" queue itself (i.e., on both positions $\mathcal{C}_{ij}+1$ and $\mathcal{C}_{ij}+2$), obtaining

$$\overline{N} = \lambda\overline{W}, \tag{9}$$

where

$$\overline{N} = 2\cdot p_{ij}(\mathcal{C}_{ij}+2) + p_{ij}(\mathcal{C}_{ij}+1), \tag{10}$$

and

$$\lambda = (\xi_{ij0}+\xi_{ij1})\cdot(\pi_{ij}+\Pi_{ij}). \tag{11}$$

Now let $\hat{N}$ be the number of tasks in the "virtual" queue found by an arriving task. The average waiting time in this queue can be obtained as follows:

$$\overline{W} = \mathbf{E}(W|\hat{N}=0)\mathbf{P}(\hat{N}=0) + \mathbf{E}(W|\hat{N}=1)\mathbf{P}(\hat{N}=1), \tag{12}$$

which can be rewritten as

$$\overline{W} = \mathcal{R}_{ij}\cdot\frac{p_{ij}(\mathcal{C}_{ij})}{p_{ij}(\mathcal{C}_{ij})+p_{ij}(\mathcal{C}_{ij}+1)} + (\tilde{\mathcal{R}}_{ij}+\mathcal{S}_{ij})\cdot\frac{p_{ij}(\mathcal{C}_{ij}+1)}{p_{ij}(\mathcal{C}_{ij})+p_{ij}(\mathcal{C}_{ij}+1)}, \tag{13}$$

where $\mathcal{R}_{ij}$ is the average remaining service time of switch–(i, j) at the time one of the outputs connected to switch–(i, j) gets blocked, given that the other output is not blocked, and $\mathcal{S}_{ij}$ is the average full service time. Thus:

$$\mathcal{R}_{ij} = \frac{\gamma_{ij}^{(1)}}{b_{ij0}} + \frac{\gamma_{ij}^{(2)}}{b_{ij1}} + \frac{\gamma_{ij}^{(3)}}{b_{ij0} + b_{ij1}}, \tag{14}$$

and $\mathcal{S}_{ij}$ is given by:

$$\begin{aligned} \mathcal{S}_{ij} \quad &= \quad \frac{\Upsilon_{ij}^{(1)}\beta_{ij0}}{b_{ij0}} + \frac{\Upsilon_{ij}^{(2)}\beta_{ij1}}{b_{ij1}} + \frac{(\Upsilon_{ij}^{(3)} + \Upsilon_{ij}^{(4)})\beta_{ij0}\beta_{ij1}}{b_{ij0} + b_{ij1}} \\ &\quad + \frac{\Upsilon_{ij}^{(5)}}{b_{ij0}} + \frac{\Upsilon_{ij}^{(6)}}{b_{ij1}} + \frac{\Upsilon_{ij}^{(7)}\beta_{ij0}}{b_{ij0} + b_{ij1}} + \frac{\Upsilon_{ij}^{(8)}\beta_{ij1}}{b_{ij0} + b_{ij1}}. \end{aligned} \tag{15}$$

After calculating $\overline{N}$ and $\overline{W}$, we can obtain probability π_{ij} using Equations 9 and 11. This enables us to finally obtain $B^*_{k,j-1,x}(s)$ by plugging the values of π_{ij} and Π_{ij} into Equation 6. We can approximate $B^*_{k,j-1,x}(s)$ by a GE distribution by matching the first two moments, $m^{(1)}_{b_{k,j-1,x}}$ and $m^{(2)}_{b_{k,j-1,x}}$. Then we have:

$$\begin{aligned} B^*_{k,j-1,x}(s) \quad &\approx \quad (1 - \beta_{k,j-1,x}) + \beta_{k,j-1,x}\frac{b_{k,j-1,x}}{s + b_{k,j-1,x}} \\ &\qquad 0 \leq k \leq L-1, \;\; 1 \leq j \leq M-1, \;\; x = 0, 1, \end{aligned} \tag{16}$$

where the approximation parameters $\mathbf{br}_{k,j-1,x} = (b_{k,j-1,x}, \beta_{k,j-1,x})$ are given by:

$$b_{k,j-1,x} \quad = \quad \frac{2m^{(1)}_{b_{k,j-1,x}}}{m^{(2)}_{b_{k,j-1,x}}}, \qquad \beta_{k,j-1,x} \quad = \quad \frac{2(m^{(1)}_{b_{k,j-1,x}})^2}{m^{(2)}_{b_{k,j-1,x}}}.$$

Now switch–$(k, j-1)$ can be analysed as a $\mathcal{S}_{2\times2}(M/GE/1/\mathcal{C}_{k,j-1} + 2)$ system by solving the balance equations given in Appendix A. We need the overall arrival rate $\lambda_{k,j-1}$ (and $\lambda'_{k,j-1}$) to switch–$(k, j-1)$ which guarantees that the throughput is $\xi_{k,j-1} = \xi_{k,j-1,0} + \xi_{k,j-1,1}$. This revised rate can be obtained from the following fixed-point problem:

$$\lambda_{k,j-1} = \frac{\xi_{k,j-1}}{1 - p^{(0)}_{k,j-1}(\mathcal{C}_{k,j-1} + 2) - p^{(0)}_{k,j-1}(\mathcal{C}_{k,j-1} + 1)[1 - \eta_{k,j-1}]}, \tag{17}$$

with

$$p^{(0)}_{k,j-1}(\mathcal{C}_{k,j-1} + 2) = \sum_{n_0=0}^{\mathcal{C}_{k,j-1}+2} p^{(0)}_{k,j-1}(n_0, \mathcal{C}_{k,j-1} + 2 - n_0).$$

If the GE distribution turns out to be improper, i.e., the GE's second parameter, $\beta_{k,j-1,x}$ (probability of not "skipping" service) is greater than 1, we set $\beta_{k,j-1,x} = 1$ and match only the first moment to get an exponential distribution. In this case, we solve a $\mathcal{S}_{2\times2}(M/M/1/\mathcal{C}_{ij} + 2)$ system. For the case that $\mathcal{C}_{ij} = 1$, equations 2 through 17 are significantly simplified. Step–0 of the approximation algorithm is summarised in Figure 7.

3.2 Step–1

This step updates the approximations of the joint queue-length distribution at the queues located in each switch of the network. The main objective of Step–1 is the distribution $p^{(1)}_{ij}(n_0, n_1)$, $0 \leq n_0 \leq \mathcal{C}_{ij} + 2$, $0 \leq n_1 \leq \mathcal{C}_{ij} + 2 - n_0$, $0 \leq i \leq L-1$, $0 \leq j \leq M-1$. As in Step–0, two units

Step–0 *(Right to left (Initialisation))*

Read input data ($\boldsymbol{\lambda}, \boldsymbol{\mathcal{C}}, \boldsymbol{\mathcal{T}}$, **R**, **S**);
Compute ξ_{ijx}, $0 \leq i \leq M-1$, $0 \leq j \leq L-1$, $x = 0, 1$;

for $i \leftarrow 0$ **to** $L-1$ **do**
begin
$B^*_{i,M-1,x}(s) \leftarrow S^*_{i,M-1,x}(s)$, $x = 0, 1$;
Analyse switch–$(i, M{-}1)$ as $\mathcal{S}_{2\times 2}(M/GE/1/\mathcal{C}_{i,M-1}{+}2)$ and *Compute* $p^{(0)}_{i,M-1}(n_0, n_1)$, $n_0 + n_1 \leq \mathcal{C}_{i,M-1} + 2$;
Compute probabilities $\Upsilon^{(k)}_{i,M-1}$, $k = 1, \ldots, 8$, $\pi_{i,M-1}$, $\Pi_{i,M-1}$, and LST's $\mathcal{G}^{(k)*}_{i,M-1}(s)$, $k = 0, 1, 2, 3$
end;

Last stage switches

for $j \leftarrow M-2$ **downto** 1 **do**
begin
for $i \leftarrow 0$ **to** $L-1$ **do**
begin
Read $\boldsymbol{\mathcal{T}}_b$ to obtain $(k, j+1)$ and $(l, j+1)$, the addresses of the switches that are connected to outputs 0 and 1 of switch–(i, j), respectively;
Obtain $B^*_{ijx}(s|\cdot)$, $x = 0, 1$;
Obtain $B^*_{ijx}(s)$, and $\mathbf{br}_{ijx}$, $x = 0, 1$;
Analyse switch–(i, j) as $\mathcal{S}_{2\times 2}(M/GE/1/\mathcal{C}_{ij}{+}2)$ and *Compute* $p^{(0)}_{ij}(n_0, n_1)$, $n_0 + n_1 \leq \mathcal{C}_{ij} + 2$;
Compute probabilities $\Upsilon^{(k)}_{ij}$, $k = 1, \ldots, 8$, π_{ij}, Π_{ij}, and LST's $\mathcal{G}^{(k)*}_{ij}(s)$, $k = 0, 1, 2, 3$
end
end;

Intermediate stages switches

for $i \leftarrow 0$ **to** $L-1$ **do**
begin
Read $\boldsymbol{\mathcal{T}}_b$ to obtain $(k, 1)$ and $(l, 1)$, the addresses of the switches that are connected to outputs 0 and 1 of switch–$(i, 0)$, respectively;
Obtain $B^*_{i,0,x}(s|\cdot)$, $x = 0, 1$;
Obtain $B^*_{i,0,x}(s)$, and $\mathbf{br}_{i,0,x}$, $x = 0, 1$;
Analyse switch–$(i, 0)$ as two $M/GE/1/\infty$ queues and *Compute* $p^{(0)}_{i,0}(n_0, n_1)$, $n_0 = 0, 1, \ldots$, and $n_1 = 0, 1, \ldots$
end;

First stage switches

Figure 7: **Step–0** *(Right to left)*

are added to the capacity of switch–(i,j) to indicate that two tasks can be blocked by switch–(i,j). In this step, the analysis is also at switch-level, i.e., the algorithm analyses each switch of the network in isolation. Unlike Step–0, the approach employed here is that initially, the *overall* arrival process (i.e., with losses) at each switch of stage–1 (second stage) is obtained and approximated by a GE interarrival time distribution, and subsequently, all switches of stage–1 are analysed as $\mathcal{S}_{2\times 2}(GE/GE/1/\mathcal{C}_{i1}+2)$ systems. The effective service time distributions are those obtained in the previous step. The algorithm proceeds in the forward (left to right) direction by determining the arrival processes to each switch in consecutive stages from stage–1 to stage–$M-1$; the processes are given (as Poisson) at stage–0.

The arrival and departure processes associated with switch–(i,j), $0 \le i \le L-1$, $0 \le j \le M-1$ are defined as follows. Let $D^*_{ijx}(s)$ be the LST of the distribution of the interdeparture time from output–x of switch–(i,j), $x = 0,1$. Let $E^*_{ijx}(s)$ be the LST of the distribution of the *effective* interarrival time at input–x of switch–(i,j), $x = 0,1$. This is just the LST for departures at the connected output in stage–$(i-1)$ $(i > 1)$ and by the given Poisson process at stage–0. As explained in Section 2.1, the incoming traffic originating from both input–0 and input–1 merges and then splits by independent probabilities (Markov routing). Therefore, a task arriving at input–x is redirected to either queue–0 or queue–1, according to the routing probabilities. Let $E^*_{ij}(s)$ be the LST of the distribution of the "superposed" *effective* interarrival times at switch–(i,j).

These LST's represent the actual arrival and departure processes observed at switch–(i,j). However, to compensate for the blocking effect and guarantee that switch–(i,j)'s throughput is ξ_{ij}, we must define *overall* arrival processes to switch–(i,j). As in Section 3.1, we model a node with blocking by a queueing system with no blocking but *losses*, i.e., arrivals to a "full" system which are rejected. The arrival rate to the non-blocking node is the overall arrival rate which is greater than that of the effective arrival process, i.e., ξ_{ij}. To analyse switch–(i,j) in isolation as a $\mathcal{S}_{2\times 2}(GE/GE/1/\mathcal{C}_{ij}+2)$ system we need the parameters of the GE approximation of the superposed overall arrival process to switch–(i,j). Let $O^*_{ij}(s)$ be the LST of the distribution of the *overall* interarrival time at switch–(i,j).

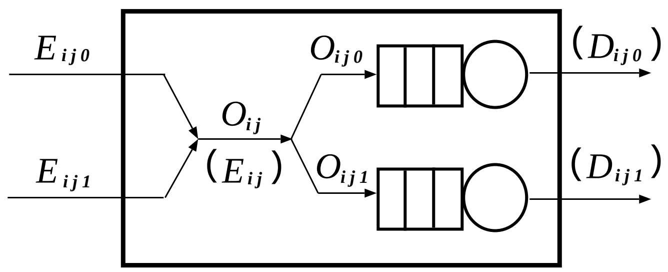

Figure 8: Switch–(i,j) – Arrival Processes.

Now, in order to obtain the LST's of the distributions of the departure processes from outputs 0 and 1 ($D^*_{ij0}(s)$ and $D^*_{ij1}(s)$, respectively), we need the LST's of the distributions of the interarrival times at these outputs. Let u_{ijx} be the probability that an arriving task at switch–(i,j) has output–x in its path (it can be calculated from the routing probabilities), and let $\mathbf{ar}_{ij} = (a_{ij}, \alpha_{ij})$ denote the parameters of the GE *overall* interarrival time distribution at switch–(i,j), $0 \le i \le L-1$, $0 \le j \le M-1$, $x = 0,1$. The arrival process whose LST is $O^*_{ij}(s)$, is split by independent probabilities, u_{ij0} and u_{ij1}, resulting in two processes with LST's $O^*_{ij0}(s)$ and $O^*_{ij1}(s)$, respectively. Thus, $O^*_{ijx}(s)$ is the LST of the overall interarrival time distribution at output–x of switch–(i,j) (after merging and splitting). As we shall see in Section 3.2.1, these distributions are used to obtain the distributions of the interdeparture times from switch–(i,j). Figure 8 depicts schematically the relationship between $E^*_{ij0}(s)$, $E^*_{ij1}(s)$, $E^*_{ij}(s)$, $O^*_{ij}(s)$, $O^*_{ij0}(s)$

and $O^*_{ij1}(s)$.

In what follows, we derive the LST of the distribution of the interdeparture time from queue–x of switch–(i,j), $D^*_{ijx}(s)$.

3.2.1 Departure Processes

We first consider the departure processes from the outputs of stage–0 switches, say output–x of switch–$(i,0)$, $0 \le i \le L-1$, $x = 0,1$. We assume that service times at these outputs are independent. If on a task's departure from queue–x at least one task is left behind in that queue, the next task comes into service and the next departure occurs at the end of its service (which may have length zero). If the departure leaves behind queue–x empty, then the next departure occurs after an arrival and its subsequent service time. Thus we have:

$$D^*_{i0x}(s) = (1-\rho_{i0x})\frac{\xi_{i0}u_{i0x}}{\xi_{i0}u_{i0x}+s}B^*_{i0x}(s) + \rho_{i0x}B^*_{i0x}(s), \tag{18}$$

where ξ_{i0} is the total external arrival rate at switch–$(i,0)$, and ρ_{i0x} is the *utilisation* of output–x of switch–$(i,0)$, which is given by:

$$\rho_{i0x}(n) = \frac{\beta_{i0x}\xi_{i0}u_{i0x}}{b_{i0x}}.$$

Now, consider the departure process from output–x of switch–(i,j), with LST $D^*_{ijx}(s)$, $0 \le i \le L-1$, $1 \le j \le M-1$, $x = 0,1$. Let δ_{ijx} be the probability that a departing task leaves queue–x of switch–(i,j) empty. The probability δ_{ijx} is given by:

$$\delta_{ijx} = \begin{cases} \displaystyle\sum_{k=0}^{c_{ij}+1} r_{ijx}(0,k), & x = 0, \\ \displaystyle\sum_{k=0}^{c_{ij}+1} r_{ijx}(k,0), & x = 1, \end{cases}$$

where $r_{ijx}(n_0,n_1)$ is the joint queue-length distribution left behind by a departing task from output–x of switch–(i,j) (see Appendix C). This is *not* the same as the steady-state distribution $p_{ijx}(n_0,n_1)$ since, for GE service time distributions, state transition need not be 1–step, as in a standard $M/G/1$ queue. However, an analogous argument applied to batches can be used. Then, $D^*_{ijx}(s)$ is given by:

$$\begin{aligned} D^*_{ijx}(s) &= \delta_{ijx}O^*_{ijx}(s)B^*_{ijx}(s) + (1-\delta_{ijx})B^*_{ijx}(s), \\ &\qquad 0 \le i \le L-1, \ 1 \le j \le M-1, \ x = 0,1. \end{aligned} \tag{19}$$

As in the case of stage–0 switches, if on a task's departure from queue–x at least one task is left behind in that queue, here with probability $(1-\delta_{ijx})$, the next task comes into service and the next departure occurs at the end of its service. If the departure leaves behind queue–x empty, then the next departure occurs after an arrival (here characterized by the LST $O^*_{ijx}(s)$) and its service time.

We now proceed to determine the arrival processes to each switch of the network.

3.2.2 Arrival Processes

Consider an arbitrary switch, say switch–(i,j), whose inputs, x and y are connected to output–z of switch–$(k,j-1)$ and input–w of switch–$(l,j-1)$, respectively, with z and w being either 0 or 1, and $0 \le (i,k,l) \le L-1$, $0 < j \le M-1$. It follows that the departure process from output–z must be the effective interarrival process at input–x. Thus, obtaining the LST's of the distributions of the interdeparture times from stage–$(j-1)$ outputs, gives us the LST's of the distributions of the interarrival times at stage–j inputs, using the respective connections (topology matrix). Then, for the case mentioned above, we have:

$$E^*_{ijx}(s) = D^*_{k,j-1,z}(s). \tag{20}$$

From $E^*_{ij0}(s)$ and $E^*_{ij1}(s)$ we can obtain the LST, $E^*_{ij}(s)$, of an approximation to the distribution of the superposed effective arrival process.

Superposition We make use of a technique of Whitt [23], which gives a renewal process approximation to the distribution of the superposition of two or more renewal processes. We call these processes superposition components. The result is given in terms of two parameters of the superposition's renewal period distribution, the mean and coefficient of variation, which are given in terms of the corresponding parameters of the superposition components. We consider the case of two components, and let them be labelled X and Y. Let ν_X be the reciprocal of the first moment of the renewal period of X (i.e., its rate) and ${c_X}^2$, its coefficient of variation. Similarly ν_Y and ${c_Y}^2$ are the parameters of process Y. The parameters ν_Z and ${c_Z}^2$, of the superposed process, are determined as follows:

$$\begin{aligned} \nu_Z &= \nu_X + \nu_Y, \\ {c_Z}^2 &= \tau\left(\frac{\nu_X {c_X}^2 + \nu_Y {c_Y}^2}{\nu_X + \nu_Y}\right) + 1 - \tau, \\ \text{where} \qquad \tau &= \frac{1}{1 + 4[1 - (\nu_X + \nu_Y)]^2}. \end{aligned}$$

Considering, as components, the processes described by $E^*_{ij0}(s)$ and $E^*_{ij1}(s)$, and using their corresponding parameters in the above equations, we obtain $\nu_{e_{ij}}$ and $c^2_{e_{ij}}$, the mean and coefficient of variation of the superposed process. These values are then used to determine the parameters of a GE distribution with same first two moments, with LST $E^*_{ij}(s)$.

Now, in order to analyse a switch in isolation, say switch–(i,j), as a $\mathcal{S}_{2\times 2}(GE/GE/1/\mathcal{C}_{ij}+2)$ system, we need the LST of the distribution of the overall interarrival times at switch–(i,j), i.e., $O^*_{ij}(s)$ which can be obtained from $E^*_{ij}(s)$. Let f_{ij} be the probability that an arriving task at either input 0 or 1, finds switch–(i,j) full and is blocked at the previous stage. Following an argument of Kuehn [12], the length of an effective interarrival interval is given by a sum of a random number K of successive overall interarrival intervals which implies that there are $K+1$ overall arrivals of which K found switch–(i,j) full with fixed probability f_{ij} and are rejected (i.e., the arrival rate to a full $\mathcal{S}_{2\times 2}(GE/GE/1/\mathcal{C}_{ij}+2)$ system is zero). The random number K is geometrically distributed with parameter f_{ij}. Assuming that the overall interarrival intervals are i.i.d., we have, by thinning the "overall" arrivals,

$$E^*_{ij}(s) = \sum_{k=0}^{\infty}(1-f_{ij}){f_{ij}}^k[O^*_{ij}(s)]^{k+1},$$

which after some algebraic manipulation gives

$$O^*_{ij}(s) = \frac{E^*_{ij}(s)}{1 - f_{ij} + f_{ij}E^*_{ij}(s)}, \quad 1 \le i \le L, \;\; 1 \le j \le M, \;\; x = 0, 1. \tag{21}$$

From the definition of f_{ij} it follows that $f_{ij} = \sum_{n=0}^{\mathcal{C}_{ij}+2} p^{(1)}_{ij}(n, \mathcal{C}_{ij}+2-n)$. Now, let $m^{(1)}_{o_{ij}}$ and $m^{(2)}_{o_{ij}}$ be the first two moments of the distribution whose LST is $O^*_{ij}(s)$. Analogously, let $m^{(1)}_{e_{ij}}$ and $m^{(2)}_{e_{ij}}$ relate to $E^*_{ij}(s)$. Then, we have:

$$m^{(1)}_{o_{ij}} = (1-f_{ij})m^{(1)}_{e_{ij}}, \qquad m^{(2)}_{o_{ij}} = (1-f_{ij})(m^{(2)}_{e_{ij}} - 2f_{ij}(m^{(1)}_{e_{ij}})^2).$$

Step–1 *(Left to right)*
for $j \leftarrow 1$ **to** $M-1$ **do**
begin
for $i \leftarrow 0$ **to** $L-1$ **do**
begin
Read $\boldsymbol{\mathcal{T}}$ to obtain $(k, j-1, z)$ and $(l, j-1, w)$,
the addresses of the outputs connected to inputs
0 and 1 of switch–(i,j), respectively;
Obtain $D^*_{k,j-1,z}(s)$ and $D^*_{l,j-1,w}(s)$;
$E^*_{ij0}(s) \leftarrow D^*_{k,j-1,z}(s)$ and $E^*_{ij1}(s) \leftarrow D^*_{l,j-1,w}(s)$;
Obtain $E^*_{ij}(s)$ (Superposition);
Obtain $O^*_{ij}(s)$;
Analyse switch–(i,j) as $\mathcal{S}_{2\times 2}(GE/GE/1/\mathcal{C}_{ij}+2)$
and *Compute* $p^{(1)}_{ij}(n_0,n_1)$, $n_0+n_1 \leq \mathcal{C}_{ij}+2$;
Obtain $O^*_{ijx}(s)$, $x = 0,1$ (Splitting)
end
end;

All switches but those of the first stage

Figure 9: **Step–1** *(Left to right)*

So, the first two moments of the distribution whose LST is $O^*_{ij}(s)$, are expressed in terms of f_{ij} (at this point unknown), and $m^{(1)}_{e_{ij}}$ and $m^{(2)}_{e_{ij}}$, obtained by superposing the processes whose LST's are $E^*_{ij0}(s)$ and $E^*_{ij1}(s)$. We use these values to determine the parameters of a GE distribution, the pair $(\lambda_{ij}, \sigma_{ij})$ say, with same first two moments. This GE distribution is then used for the interarrival time distribution when solving the $\mathcal{S}_{2\times 2}(GE/GE/1/\mathcal{C}_{ij}+2)$ system for switch–(i,j). If the GE distribution turns out to be improper, we set $\theta_{ij} = 1$ and match only the first moment to get an exponential distribution, as in section 3.1. In this case, we obtain switch–(i,j)'s steady-state joint queue length distribution by solving the corresponding $\mathcal{S}_{2\times 2}(M/GE/1/\mathcal{C}_{ij}+2)$ system. The value of f_{ij} can be found by solving the following fixed-point problem:

$$\lambda_{ij} = \frac{\sigma_{ij}\xi_{ij}}{1 - p^{(1)}_{ij}(\mathcal{C}_{ij}+2) - p^{(1)}_{ij}(\mathcal{C}_{ij}+1)[1-\eta_{ij}\sigma_{ij}]}$$

which guarantees that the throughput at switch–(i,j) is ξ_{ij}.

Splitting Finally, we obtain the parameters of the GE approximations of $O^*_{ijx}(s)$ using the "splitting" probabilities u_{ij0} and u_{ij1}. Let $\nu_{o_{ij}}$ and $c^2_{o_{ij}}$ be the rate and squared coefficient of variation of the overall superposed arrival process. The results are also given in terms of mean and squared coefficient of variation, for $x = 0,1$, as follows (see [23]):

$$\nu_{o_{ijx}} = u_{ijx}\nu_{o_{ij}}, \qquad c^2_{o_{ijx}} = u_{ijx}c^2_{o_{ij}} + 1 - u_{ijx}.$$

Given a squared coefficient of variation c^2, the corresponding second parameter of a GE distribution, θ, is given by $\theta = 2/(c^2+1)$. Step–1 of the approximation algorithm is summarised in Figure 9.

3.3 Step–2

This step has the same structure as Step–0, i.e., it analyses the network at the switch-level, moving backwards from stage–$(M-1)$ to stage–0. Note that, since stage–$(M-1)$ switches cannot be blocked, Step–2 starts from stage–$(M-2)$. As in Step–0, the effective service time distribution at each queue of each switch is calculated, but unlike Step–0, the arrival process at each input pin is approximated by a GE distribution. It then follows that we can not make use of the PASTA argument in order to calculate the probabilities $\gamma_{ij}^{(1)}$, $\gamma_{ij}^{(2)}$, $\gamma_{ij}^{(3)}$, and $\Upsilon_{ij}^{(k)}$, $k=1,\ldots,8$, since they are all conditioned upon arrival instants. Therefore, we must obtain the joint queue-length probability distribution at these time epochs. But the fact that the interarrival times are GE–distributed implies that the joint queue-length distribution seen by an arriving batch, irrespective of the batch's size, is actually the *equilibrium* distribution, since batches's interarrival instants are exponentially distributed.

In order to obtain the two parameters of the GE approximation of the *overall* interarrival time distribution we assume that the superposed effective arrival process at switch–(i,j) obtained in Step–1 (with LST, $E_{ij}^*(s)$) does not change in Step–2. Note that, as in the previous two steps, the re-calculated overall arrival rate to switch–(i,j) guarantees that the throughput is equal to ξ_{ij}. It follows that each switch is analysed as a $\mathcal{S}_{2\times 2}(GE/GE/1/\mathcal{C}+2)$ system where again two units are added to the switch capacity in order to take the blocking effect into account. As a result, Step–2 produces the joint queue-length distributions $p_{ij}^{(2)}(n_0,n_1)$, $1\leq i\leq L$, $1\leq j\leq M$, $n_0+n_1\leq \mathcal{C}_{ij}+2$, which is compared to the distribution obtained in Step–1, $p_{ij}^{(1)}(n_0,n_1)$, in order to determine the convergence of the algorithm. If necessary, the algorithm returns to Step–1 this time using the effective service time distribution obtained in Step–2. Steps 1 and 2 are repeated until convergence is achieved according to the maximum difference between the probability distributions $p_{ij}(n_0,n_1)$ obtained in successive steps, i.e.

Step–0;
repeat
  Step–1;
  Step–2;
until $max_{\substack{n_0,n_1\\(n_0+n_1\leq \mathcal{C}_{ij})}} |p_{ij}^{(1)}(n_0,n_1)-p_{ij}^{(2)}(n_0,n_1)|<\varepsilon,$
  $0\leq i\leq L-1,\ 0\leq j\leq M-1.$

The main result of the proposed algorithm is the distribution $p_{ij}(n)$, $0\leq n\leq \mathcal{C}_{ij}+2$. When the algorithm stops, the actual probability of switch–(i,j) being full (blocking probability), $p_{ij}(\mathcal{C}_{ij})$, is given by $\sum_{l=0}^{2} p_{ij}(\mathcal{C}_{ij}+l)$. We can finally obtain the mean queue-length at each switch using the distribution $p_{ij}(n)$, $0\leq n\leq \mathcal{C}_{ij}$, for $0\leq i\leq L-1$, $1\leq j\leq M-1$. Note that $p_{ij}(\mathcal{C}_{ij}+1)$ simply gives the probability that switch–(i,j) is full <u>and</u> one of the outputs connected to it is blocked. The position $\mathcal{C}_{ij}+1$ is fictitious in the sense that it is not considered as an extra position of switch–(i,j) but rather as a representation of a particular situation. Analogously, $p_{ij}(\mathcal{C}_{ij}+2)$ simply gives the probability that switch–(i,j) is full <u>and</u> both outputs connected to it are blocked.

We have not investigated formal conditions for convergence nor of the existence and uniqueness of the resulting fixed-point. We conjecture that the algorithm would not converge for an unstable system. Step–2 of the approximation algorithm is summarised in Figure 10.

4 Numerical Results

In this section we give some numerical results of the analytical model and compare these with simulation; see Tables 1 – 10. We consider an 8×8 Banyan network with the same topology as that shown in Figure 1. For Tables 1 – 4, each output pin has a GE-distributed service time with parameters $\mathbf{br}=(0.5,0.5)$. These values were chosen (partly) so that the average service time at each output equals 1. The external arrival rate at each input pin is $\lambda_i=0.1$, $i=0,\ldots,7$, so that the network total arrival rate equals 0.8 and all tasks are stochastically identical. Thus the network is stable since even if all traffic goes to a single output (the hot-spot), the total service rate in the final stage is 1 which is greater than the total arrival rate. The top output (hot-spot)

Step–2 *(Right to left)*

for $j \leftarrow M-2$ **downto** 1 **do**
begin
  for $i \leftarrow 0$ **to** $L-1$ **do**
  begin
    Read $\boldsymbol{\mathcal{T}}_b$ to obtain $(k, j+1)$ and $(l, j+1)$, the addresses of the switches that are connected to outputs 0 and 1 of switch–(i,j), respectively;
    Obtain $B^*_{ijx}(s|\cdot)$, $x = 0,1$;
    Obtain $B^*_{ijx}(s)$, and $\mathbf{br}_{ijx}$, $x = 0,1$;
    Analyse switch–(i,j) as $\mathcal{S}_{2\times 2}(GE/GE/1/\mathcal{C}_{ij}+2)$ and *Compute* $p^{(2)}_{ij}(n_0,n_1)$, $n_0+n_1 \leq \mathcal{C}_{ij}+2$;
    Compute probabilities $\Upsilon^{(k)}_{ij}$, $k = 1,\ldots,8$, π_{ij}, Π_{ij}, and the LST's $\mathcal{G}^{(k)*}_{ij}(s)$, $k = 0,1,2,3$
  end
end;

Intermediate stages switches

for $i \leftarrow 0$ **to** $L-1$ **do**
begin
  Read $\boldsymbol{\mathcal{T}}_b$ to obtain $(k,1)$ and $(l,1)$, the addresses of the switches that are connected to outputs 0 and 1 of switch–$(i,0)$, respectively;
  Obtain $B^*_{i,0,x}(s|\cdot)$, $x = 0,1$;
  Obtain $B^*_{i,0,x}(s)$, and $\mathbf{br}_{i,0,x}$, $x = 0,1$;
  Analyse switch–$(i,0)$ as two $M/GE/1/\infty$ queues and *Compute* $p^{(2)}_{i,0}(n_0,n_1)$, $n_0 = 0,1,\ldots$, and $n_1 = 0,1,\ldots$
end;

First stage switches

Figure 10: **Step–2** *(Right to left)*

is selected with the same probability, r_{i0}, by all tasks and all other outputs are selected with equal probability. For Tables 1 – 2, the routing probabilities r_{i0}, $0 \le i \le 7$, are equal to 0.23, while the remaining probabilities, r_{ij}, for $1 \le i \le 7$ and $0 \le j \le 7$, are all equal to 0.11. For Tables 3 – 4, the routing probabilities r_{i0}, $0 \le i \le 7$, are equal to 0.86, while the remaining probabilities, r_{ij}, for $1 \le i \le 7$ and $0 \le j \le 7$, are all equal to 0.02. This is therefore an extreme hot-spot but the predictions retain reasonable accuracy. The back-propagation effect of the hot-spot can be seen clearly, both in the blocking probabilities and the mean queue lengths. Buffer occupancy is significantly higher throughout the decode tree of the hot-spot, the difference compared with buffers in paths to cool outputs increasing at later stages. Other performance measures of interest follow directly from these results, for example, mean switch sojourn time, by application of Little's result, and hence mean transmission time. As pointed out in Section 3, the effective arrival rate at each input pin of any stage can be directly obtained from the external arrival rates, $\boldsymbol{\lambda}$ and from the network's parameters, $\boldsymbol{\mathcal{T}}$ and $\mathbf{R}$, since stage–0 switches have infinite buffers and therefore, every task arrival is accepted into the network (i.e., there are no losses). For a given switch, say switch–(i, j), the total arrival rate is given by the sum of the effective arrival rates at both outputs 0 and 1. The mean queue-length at switch–(i, j) can be obtained from the results of the approximation algorithm as indicated in Section 3.3. Thus, the mean sojourn time at switch–(i, j) can be evaluated by direct application of Little's result. The same reasoning can be applied to obtain the mean transmission time in a given path. Tables 1 – 10 give the network's mean latency time which is obtained by applying Little's result as follows: the mean number of tasks in the whole network is calculated by summing over all switches and this value is then divided by the network's total arrival rate. The simulations verify the accuracy of the analytical results.

Tables 5 – 8 give results for a non-uniform traffic pattern characterized by the routing probability matrix **R'**, given below:

$$\mathbf{R'} = \begin{bmatrix} 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.2 & 0.2 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.2 & 0.2 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.26 & 0.26 & 0.04 & 0.04 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.26 & 0.26 & 0.04 & 0.04 \\ 0.125 & 0.125 & 0.125 & 0.125 & 0.125 & 0.125 & 0.125 & 0.125 \\ 0.125 & 0.125 & 0.125 & 0.125 & 0.125 & 0.125 & 0.125 & 0.125 \\ 0.125 & 0.125 & 0.125 & 0.125 & 0.125 & 0.125 & 0.125 & 0.125 \\ 0.125 & 0.125 & 0.125 & 0.125 & 0.125 & 0.125 & 0.125 & 0.125 \end{bmatrix}$$

These values where chosen so that the traffic flows arriving on distinct inputs at a switch (in this particular case, switch–$(1, 1)$) have routing bias towards distinct switch outputs. Considering the topology of the network shown in Figure 1, the traffic flow arriving at input–0 of switch–$(1, 1)$ is given by $\lambda_0(r_{04}+r_{05}+r_{06}+r_{07})+\lambda_1(r_{14}+r_{15}+r_{16}+r_{17})$, where $\lambda_0(r_{04}+r_{05})+\lambda_1(r_{14}+r_{15})$ is the traffic flow from input–0 to output–0, and $\lambda_0(r_{06} + r_{07}) + \lambda_1(r_{16} + r_{17})$ is the traffic flow from input–0 to output–1. According to the routing matrix **R'**, given above, these traffic flows are then respectively $0.2(\lambda_0 + \lambda_1)$ and $0.4(\lambda_0 + \lambda_1)$. Analogously, the traffic flow arriving at input–1 of switch–$(1, 1)$ is given by $\lambda_2(r_{24} + r_{25} + r_{26} + r_{27}) + \lambda_3(r_{34} + r_{35} + r_{36} + r_{37})$, where $\lambda_2(r_{24} + r_{25}) + \lambda_3(r_{34} + r_{35})$ is the traffic flow from input–1 to output–0, and $\lambda_2(r_{26} + r_{27}) + \lambda_3(r_{36} + r_{37})$ is the traffic flow from input–1 to output–1. Again, according to **R'**, we have a rate $0.52(\lambda_2 + \lambda_3)$ from 1 to 0 and $0.08(\lambda_2 + \lambda_3)$, from 1 to 1. For the case of Tables 5 and 6 the external arrival rate at each input pin is equal to 0.1, whereas in Tables 7 and 8 the arrival rates are given by $\boldsymbol{\lambda} = [0.02, 0.02, 0.2, 0.2, 0.02, 0.02, 0.2, 0.2]$. Again, looking at switch–$(1, 1)$, it can be noticed that there is much more traffic arriving at input–0, 0.24 in this case, than at input–1, which is 0.024. The simulation indicates that the analytical model performs reasonably well even for these cases where there is the combined effect of unbalanced traffic and routing bias. The results are slightly pessimistic for the switches most affected by this combined effect.

Level		*Stage*					
		0		1		2	
		Appr.	Sim.	Appr.	Sim.	Appr.	Sim.
0	BP	—	—	0.049	0.027 ± 0.003	0.073	0.041 ± 0.002
	MQL	0.215	0.204 ± 0.005	0.269	0.245 ± 0.005	0.325	0.284 ± 0.015
1	BP	—	—	0.035	0.015 ± 0.002	0.034	0.015 ± 0.001
	MQL	0.215	0.198 ± 0.006	0.205	0.173 ± 0.008	0.192	0.172 ± 0.005
2	BP	—	—	0.049	0.027 ± 0.001	0.034	0.014 ± 0.002
	MQL	0.215	0.202 ± 0.003	0.269	0.234 ± 0.003	0.192	0.169 ± 0.008
3	BP	—	—	0.035	0.015 ± 0.002	0.034	0.015 ± 0.001
	MQL	0.215	0.203 ± 0.007	0.205	0.182 ± 0.003	0.192	0.175 ± 0.004

Table 1: Blocking probabilities (BP) and mean queue lengths (MQL). Switches located at stages 1 and 2 have buffers of size 2. Routing probabilities $r_{i0} = 0.23$, $i = 0, \ldots, 7$. Network's mean latency time = 3.38 (simul., 3.15 ± 0.08).

A probable source of this inaccuracy in our model lies in the handling of the two extra, artificial slots added to a switch's buffer to account for blocking. The analysis of the queueing system in a given $S_{2\times 2}$ node assumes a *single* arrival process and the extra slots are only represented via a reduced arrival rate (λ') when one slot is occupied, corresponding to only one non-blocked input (with zero arrival rate when both slots are occupied, of course). This is an aggregating approximation since no account is taken of *which* input is blocked. In the uniform case, of course, this does not matter, which is why the model then performs much better.

In Tables 5 – 10, the network outputs are slower than all internal switches, which in practice is often the case. To produce such effect, each output pin has a GE-distributed service time with parameters $\mathbf{br} = (0.5, 0.7)$ resulting in an average service time of 1.4 as opposed to 1 for all the internal switches.

Tables 9 and 10 give results for heavy traffic. For Table 9, the external arrival rate at each input is 0.4 resulting in a total arrival rate of 3.2. For Table 10, the external arrival rate at each input is 0.5 resulting in a total arrival rate of 4. The results given in those two tables show that the model gives pessimistic results for both mean queue-length and blocking probability.

The simulation results are shown with 95% confidence intervals. The model shows much better agreement with simulation for mean queue lengths than for blocking probabilities where some errors were around 100% on very small values. However, in this case the simulation is less reliable since so few tasks get blocked and no special techniques were used to handle these "rare events". The confidence bands of the simulation estimates are therefore very wide and those given must be regarded with some suspicion.

The stability condition for an internal node in general is non-trivial to obtain since its service rate may be constrained by downstream blocking in an unknown way. However, for final stage switches which cannot be blocked, the stability condition for an output is just that the effective arrival rate must be less than the given service rate.

5 Conclusions

We have presented a novel algorithm to predict the performance of asynchronous, packet-switched banyan networks with finite buffers in the crossbars, i.e., with blocking and general service times. We assume that first stage switches have infinite buffers with Poisson arrivals and that final stage switches are not blocked, but our algorithm can be easily extended to cases

Level		Stage					
		0		1		2	
		Appr.	Sim.	Appr.	Sim.	Appr.	Sim.
0	BP	—	—	2.427	2.212 ± 1.108	7.641	8.540 ± 2.542
	MQL	0.201	0.195 ± 0.021	0.235	0.222 ± 0.036	0.368	0.282 ± 0.102
1	BP	—	—	0.532	0.460 ± 0.351	0.624	0.564 ± 0.442
	MQL	0.201	0.189 ± 0.043	0.182	0.171 ± 0.013	0.210	0.179 ± 0.024
2	BP	—	—	2.427	1.572 ± 1.315	0.624	0.418 ± 0.324
	MQL	0.201	0.192 ± 0.036	0.235	0.227 ± 0.018	0.210	0.182 ± 0.031
3	BP	—	—	0.532	0.668 ± 0.217	0.624	0.735 ± 0.230
	MQL	0.201	0.193 ± 0.035	0.182	0.173 ± 0.016	0.210	0.185 ± 0.022

Table 2: Blocking probabilities (BP) ($\times 10^{-4}$) and mean queue lengths (MQL). Switches located at stages 1 and 2 have buffers of size 4. Routing probabilities $r_{i0} = 0.23$, $i = 0, \ldots, 7$. Network's mean latency time = 3.30 (simul., 2.97 ± 0.15).

Level		Stage					
		0		1		2	
		Appr.	Sim.	Appr.	Sim.	Appr.	Sim.
0	BP	—	—	0.1998	0.2450 ± 0.0175	0.5411	0.4107 ± 0.0104
	MQL	0.341	0.359 ± 0.011	0.594	0.715 ± 0.009	1.230	1.021 ± 0.025
1	BP	—	—	0.0012	0.0005 ± 0.0001	0.0012	0.0005 ± 0.0001
	MQL	0.341	0.360 ± 0.009	0.033	0.032 ± 0.002	0.032	0.031 ± 0.003
2	BP	—	—	0.1998	0.2461 ± 0.0099	0.0012	0.0005 ± 0.0003
	MQL	0.341	0.354 ± 0.004	0.594	0.720 ± 0.018	0.032	0.029 ± 0.002
3	BP	—	—	0.0012	0.0005 ± 0.0001	0.0012	0.0005 ± 0.0002
	MQL	0.341	0.358 ± 0.008	0.033	0.031 ± 0.001	0.032	0.031 ± 0.002

Table 3: Blocking probabilities (BP) and mean queue lengths (MQL). Switches located at stages 1 and 2 have buffers of size 2. Routing probabilities $r_{i0} = 0.86$, $i = 0, \ldots, 7$. Network's mean latency time = 4.93 (simul., 5.03 ± 0.17).

Level		Stage					
		0		1		2	
		Appr.	Sim.	Appr.	Sim.	Appr.	Sim.
0	BP	—	—	0.030	0.030 ± 0.006	0.175	0.175 ± 0.0018
	MQL	0.245	0.241 ± 0.012	0.717	0.692 ± 0.009	1.994	1.583 ± 0.025
1	BP	—	—	0.	0.	0.	0.
	MQL	0.245	0.242 ± 0.010	0.035	0.032 ± 0.003	0.033	0.032 ± 0.002
2	BP	—	—	0.032	0.032 ± 0.008	0.	0.
	MQL	0.245	0.241 ± 0.008	0.717	0.709 ± 0.011	0.033	0.032 ± 0.002
3	BP	—	—	0.	0.	0.	0.
	MQL	0.245	0.239 ± 0.012	0.035	0.031 ± 0.003	0.033	0.032 ± 0.003

Table 4: Blocking probabilities (BP) and mean queue lengths (MQL). Switches located at stages 1 and 2 have buffers of size 4. Routing probabilities $r_{i0} = 0.86$, $i = 0, \ldots, 7$. Network's mean latency time = 5.72 (simul., 5.43 ± 0.12).

Level		Stage					
		0		1		2	
		Appr.	Sim.	Appr.	Sim.	Appr.	Sim.
0	BP	—	—	0.031	0.012 ± 0.002	0.053	0.022 ± 0.002
	MQL	0.225	0.201 ± 0.012	0.195	0.161 ± 0.005	0.291	0.246 ± 0.008
1	BP	—	—	0.070	0.031 ± 0.002	0.053	0.022 ± 0.001
	MQL	0.225	0.200 ± 0.009	0.319	0.249 ± 0.005	0.291	0.246 ± 0.004
2	BP	—	—	0.047	0.021 ± 0.008	0.094	0.049 ± 0.003
	MQL	0.225	0.202 ± 0.007	0.249	0.207 ± 0.004	0.409	0.337 ± 0.005
3	BP	—	—	0.049	0.019 ± 0.002	0.063	0.026 ± 0.002
	MQL	0.225	0.201 ± 0.011	0.257	0.205 ± 0.007	0.320	0.268 ± 0.011

Table 5: Blocking probabilities (BP) and mean queue lengths (MQL). Switches located at stages 1 and 2 have buffers of size 2. Routing probabilities given by **R'**. Network's mean latency time = 4.04 (simul., 3.40 ± 0.23).

Level		Stage 0 Appr.	Stage 0 Sim.	Stage 1 Appr.	Stage 1 Sim.	Stage 2 Appr.	Stage 2 Sim.
0	BP	—	—	0.085	0.025 ± 0.002	0.231	0.018 ± 0.015
	MQL	0.219	0.210 ± 0.006	0.189	0.161 ± 0.005	0.303	0.250 ± 0.015
1	BP	—	—	0.432	0.031 ± 0.02	0.233	0.009 ± 0.008
	MQL	0.219	0.208 ± 0.008	0.312	0.241 ± 0.013	0.303	0.247 ± 0.003
2	BP	—	—	0.201	0.032 ± 0.008	0.708	0.045 ± 0.012
	MQL	0.219	0.211 ± 0.004	0.246	0.199 ± 0.006	0.440	0.339 ± 0.003
3	BP	—	—	0.205	0.006 ± 0.005	0.318	0.004 ± 0.004
	MQL	0.219	0.202 ± 0.011	0.247	0.200 ± 0.008	0.336	0.268 ± 0.007

Table 6: Blocking probabilities (BP) ($\times 10^{-2}$) and mean queue lengths (MQL). Switches located at stages 1 and 2 have buffers of size 4. Routing probabilities given by **R'**. Network's mean latency time = 4.06 (simul., 3.42 ± 0.18).

Level		Stage 0 Appr.	Stage 0 Sim.	Stage 1 Appr.	Stage 1 Sim.	Stage 2 Appr.	Stage 2 Sim.
0	BP	—	—	0.030	0.015 ± 0.001	0.049	0.026 ± 0.003
	MQL	0.048	0.045 ± 0.008	0.204	0.185 ± 0.004	0.294	0.274 ± 0.009
1	BP	—	—	0.108	0.054 ± 0.007	0.064	0.026 ± 0.004
	MQL	0.723	0.528 ± 0.012	0.405	0.315 ± 0.013	0.323	0.269 ± 0.003
2	BP	—	—	0.056	0.024 ± 0.008	0.160	0.068 ± 0.004
	MQL	0.045	0.042 ± 0.002	0.280	0.228 ± 0.004	0.573	0.445 ± 0.009
3	BP	—	—	0.061	0.025 ± 0.001	0.042	0.017 ± 0.002
	MQL	0.657	0.508 ± 0.010	0.294	0.236 ± 0.005	0.251	0.218 ± 0.003

Table 7: Blocking probabilities (BP) and mean queue lengths (MQL). Switches located at stages 1 and 2 have buffers of size 2. Routing probabilities given by **R'**. Network's mean latency time = 4.66 (simul., 3.75 ± 0.19).

Level		Stage 0 Appr.	Stage 0 Sim.	Stage 1 Appr.	Stage 1 Sim.	Stage 2 Appr.	Stage 2 Sim.
0	BP	—	—	0.132	0.033 ± 0.004	0.311	0.021 ± 0.008
	MQL	0.042	0.045 ± 0.002	0.213	0.177 ± 0.005	0.340	0.273 ± 0.014
1	BP	—	—	1.001	0.117 ± 0.011	0.306	0.012 ± 0.009
	MQL	0.621	0.471 ± 0.009	0.382	0.303 ± 0.012	0.340	0.273 ± 0.003
2	BP	—	—	0.321	0.013 ± 0.012	2.04	0.811 ± 0.013
	MQL	0.042	0.043 ± 0.002	0.280	0.226 ± 0.004	0.641	0.462 ± 0.003
3	BP	—	—	0.321	0.011 ± 0.008	0.018	0.009 ± 0.005
	MQL	0.602	0.485 ± 0.007	0.283	0.223 ± 0.005	0.260	0.218 ± 0.005

Table 8: Blocking probabilities (BP) ($\times 10^{-2}$) and mean queue lengths (MQL). Switches located at stages 1 and 2 have buffers of size 4. Routing probabilities given by **R'**. Network's mean latency time = 4.60 (simul., 3.64 ± 0.15).

Level		Stage 0 Appr.	Stage 0 Sim.	Stage 1 Appr.	Stage 1 Sim.	Stage 2 Appr.	Stage 2 Sim.
0	BP	—	—	0.879	0.588 ± 0.019	0.645	0.384 ± 0.018
	MQL	85.12	44.23 ± 3.53	1.935	1.401 ± 0.020	1.694	1.166 ± 0.037
1	BP	—	—	0.879	0.583 ± 0.035	0.645	0.380 ± 0.010
	MQL	85.12	42.52 ± 2.67	1.935	1.392 ± 0.035	1.694	1.160 ± 0.021
2	BP	—	—	0.879	0.591 ± 0.023	0.645	0.376 ± 0.026
	MQL	85.12	41.61 ± 2.37	1.935	1.403 ± 0.029	1.694	1.157 ± 0.041
3	BP	—	—	0.879	0.583 ± 0.032	0.645	0.381 ± 0.017
	MQL	85.12	43.37 ± 3.72	1.935	1.395 ± 0.038	1.694	1.163 ± 0.026

Table 9: Blocking probabilities (BP) and mean queue lengths (MQL). Switches located at stages 1 and 2 have buffers of size 2. Uniform routing pattern.

Level		Stage 0 Appr.	Stage 0 Sim.	Stage 1 Appr.	Stage 1 Sim.	Stage 2 Appr.	Stage 2 Sim.
0	BP	—	—	0.761	0.274 ± 0.034	0.456	0.150 ± 0.020
	MQL	28.31	9.32 ± 0.94	3.561	2.370 ± 0.137	2.994	2.231 ± 0.123
1	BP	—	—	0.761	0.272 ± 0.019	0.456	0.148 ± 0.021
	MQL	28.31	8.77 ± 1.56	3.561	2.396 ± 0.041	2.994	2.208 ± 0.090
2	BP	—	—	0.761	0.281 ± 0.059	0.456	0.149 ± 0.015
	MQL	28.31	8.97 ± 0.99	3.561	2.426 ± 0.202	2.994	2.218 ± 0.066
3	BP	—	—	0.761	0.272 ± 0.018	0.456	0.145 ± 0.017
	MQL	28.31	8.50 ± 1.21	3.561	2.378 ± 0.055	2.994	2.221 ± 0.062

Table 10: Blocking probabilities (BP) and mean queue lengths (MQL). Switches located at stages 1 and 2 have buffers of size 4. Uniform routing pattern.

where these constrains do not hold. The approximation procedure is based on a known overall approach which has been used effectively to model blocking in tandem networks of queues. However, it handles more complex dependencies between switches arising from packet routings as well as blocking at downstream crossbars.

The accuracy of the model relative to simulation is good, especially at low-to-medium traffic, and the hot-spot phenomenon is predicted well. There are two immediate ways in which the accuracy of the single node models could be improved, at the expense of greater computational complexity. First, the arrival processes at each input could be represented separately, avoiding the approximation involved in superimposing and splitting. However, the balance equations would clearly become much more complicated with the addition of a new arrival process. Secondly, the dependence between different nodes at stage–$(j-1)$ connected to the same node at stage–j could be included in the model using the following observation. In many banyan topologies, if two switches in stage–$(j-1)$ are neighbours (due to an output from each being connected to the inputs of the same switch in stage–j), then they are also *neighbours* due to their other outputs being connected to the same switch in stage–j (different to the previous one). Thus, a more precise description of blocking could be obtained by considering pairs of neighbour-switches in each stage, rather than just single switches as we have done. The price to pay would be the analysis of two $\mathcal{S}_{2\times2}(GE/GE/1/\mathcal{C})$ systems jointly, with a state space size the square of that for a single such system.

As we noted in the previous section, a limitation of the model in non-uniform traffic is caused by our representation of the artificial slots added to each blocking node. We could improve this part of the model by considering both arrival processes to the two inputs separately, at somewhat greater computational expense. It would only be necessary to extend the state space by tagging every state with node-population $\mathcal{C}-1$ (in the notation of Appendices A and B, i.e. when one extra slot is occupied) to indicate which one of the two inputs is blocked. A superposed arrival process would probably be adequate at populations less than $\mathcal{C}-2$ but arrivals at population $\mathcal{C}-2$ would have to be distinguished, each input leading to a different tagged state. The arrival rate in each tagged state would then be that of the non-blocked input given by the tag. Complementary to this, more efficient solution methods should be sought for the single (or double) node building blocks, and one possibility would be to use maximum entropy methods [9], especially since we are already using GE distributions.

On a more theoretical level, we have observed that necessary and sufficient conditions for stability of the whole network are not obvious, due to the limiting effect on a node's service rate of downstream blocking. Essentially, the stability condition for a single output is that its arrival rate be less that its effective service rate, which we can only approximate. This is the subject of further research.

A Balance Equations of a Switch with Poisson Arrivals

The complete set of balance equations for the $\mathcal{S}_{2\times2}(M/GE/1/\mathcal{C})$ system described in Section 2.1 is given below. The arrival rate of the Poisson process is λ at switch populations less than $\mathcal{C}-1$, λ' at population $\mathcal{C}-1$ and 0 at population $\mathcal{C}$. The rate λ corresponds to both inputs to an actual switch in the network (before superposition) being active and λ' to only one of them, the other being blocked by virtue of the fact that one of the two extra buffer slots is occupied when the population is $\mathcal{C}-1$.

We first define the functions, $h_a(x)$ and I_x as:

$$h_a(x) = \begin{cases} a & \text{if } x = 0, \\ 1 & \text{otherwise,} \end{cases} \qquad I_x = \begin{cases} 1 & \text{if } x \text{ is True,} \\ 0 & \text{otherwise.} \end{cases}$$

The balance equations are then as follows:

$$p(0,0)[\lambda v\theta_1 + \lambda u\theta_0] = \mu_0 \sum_{k=1}^{\mathcal{C}} p(k,0)(1-\theta_0)^{k-1} + \mu_1 \sum_{k=1}^{\mathcal{C}} p(0,k)(1-\theta_1)^{k-1}, \tag{22}$$

$$\begin{aligned} p(0,j)[\lambda v + \lambda u\theta_0 + \mu_1] &= p(0,j-1)\lambda v h_{\theta_1}(j-1) \\ &+ \mu_0 \sum_{k=1}^{\mathcal{C}-j} p(k,j)(1-\theta_0)^{k-1} + \mu_1\theta_1 \sum_{k=j+1}^{\mathcal{C}} p(0,k)(1-\theta_1)^{k-j-1}, \\ &1 \le j \le \mathcal{C}-1, \end{aligned} \tag{23}$$

$$\begin{aligned} p(i,0)[\lambda u + \lambda v\theta_1 + \mu_0] &= p(i-1,0)\lambda u h_{\theta_0}(i-1) \\ &+ \mu_0\theta_0 \sum_{k=i+1}^{\mathcal{C}} p(k,0)(1-\theta_0)^{k-i-1} + \mu_1 \sum_{k=1}^{\mathcal{C}-i} p(i,k)(1-\theta_1)^{k-1}, \\ &1 \le i \le \mathcal{C}-1, \end{aligned} \tag{24}$$

$$\begin{aligned} p(i,j)[\lambda + \mu_0 + \mu_1] &= p(i-1,j)\lambda u h_{\theta_0}(i-1) + p(i,j-1)\lambda v h_{\theta_1}(j-1) \\ &+\mu_0\theta_0 \sum_{k=i+1}^{\mathcal{C}-j} p(k,j)(1-\theta_0)^{k-i-1} + \mu_1\theta_1 \sum_{k=j+1}^{\mathcal{C}-i} p(i,k)(1-\theta_1)^{k-j-1}, \\ &1 \le i \le \mathcal{C}-2, \;\; 1 \le j \le \mathcal{C}-1-i, \end{aligned} \tag{25}$$

$$p(\mathcal{C},0)\mu_0 = p(\mathcal{C}-1,0)\lambda' u, \tag{26}$$

$$p(0,\mathcal{C})\mu_1 = p(0,\mathcal{C}-1)\lambda' v, \tag{27}$$

$$\begin{aligned} p(i,\mathcal{C}-i)[\mu_0 + \mu_1] &= p(i-1,\mathcal{C}-i)\lambda' u h_{\theta_0}(i-1) \\ &+p(i,\mathcal{C}-i-1)\lambda' v h_{\theta_1}(\mathcal{C}-i-1), \\ &1 \le i \le \mathcal{C}-1. \end{aligned} \tag{28}$$

B Balance Equations of a Switch with GE Interarrival Times

The complete set of balance equations for the $\mathcal{S}_{2\times 2}(GE/GE/1/\mathcal{C})$ system described in Section 2.1 is given below. The parameter of the exponential stage of the GE interarrival time distribution is λ at switch populations less than $\mathcal{C}-1$, λ' at $\mathcal{C}-1$ and 0 at $\mathcal{C}$; cf. the case of Poisson arrivals. Moreover, it is also assumed that at any arrival instant, no task enters the switch by by-passing the exponential delay if this would lead to a population greater than $\mathcal{C}$. The balance equations are similar to those obtained for Poisson arrivals, but with the extra complexity introduced by (geometric) bulk arrivals taken into account; batches arise from tasks skipping exponential delays. The resulting equations are as follows:

$$\begin{aligned} p(0,0)\lambda[1 - \frac{u\sigma(1-\theta_0)}{\theta_0 + \sigma - \sigma\theta_0} - \frac{v\sigma(1-\theta_1)}{\theta_1 + \sigma - \sigma\theta_1}] &= \\ \mu_0 \sum_{k=1}^{\mathcal{C}} p(k,0)(1-\theta_0)^{k-1} + \mu_1 \sum_{k=1}^{\mathcal{C}} p(0,k)(1-\theta_1)^{k-1} \end{aligned} \tag{29}$$

$$p(0,j)[\mu_1 + (\lambda\, I_{j<\mathcal{C}-1} + \lambda'\, I_{j=\mathcal{C}-1})(1 - \frac{u\sigma(1-\theta_0)}{\theta_0+\sigma-\sigma\theta_0})] =$$
$$\frac{p(0,0)\lambda\sigma v\theta_1(1-\sigma)^{j-1}}{\theta_1+\sigma-\sigma\theta_1} + \lambda\sigma v\sum_{k=1}^{j-1} p(0,j-k)(1-\sigma)^{k-1}$$
$$+ \mu_0\sum_{k=1}^{\mathcal{C}-j} p(k,j)(1-\theta_0)^{k-1} + \mu_1\theta_1\sum_{k=1}^{\mathcal{C}-j} p(0,j+k)(1-\theta_1)^{k-1},$$
$$1 \le j \le \mathcal{C}-1, \tag{30}$$

$$p(i,0)[\mu_0 + (\lambda\, I_{i<\mathcal{C}-1} + \lambda'\, I_{i=\mathcal{C}-1})(1 - \frac{v\sigma(1-\theta_1)}{\theta_1+\sigma-\sigma\theta_1})] =$$
$$\frac{p(0,0)\lambda\sigma u\theta_0(1-\sigma)^{i-1}}{\theta_0+\sigma-\sigma\theta_0} + \lambda\sigma u\sum_{k=1}^{i-1} p(i-k,0)(1-\sigma)^{k-1}$$
$$+ \mu_1\sum_{k=1}^{\mathcal{C}-i} p(i,k)(1-\theta_1)^{k-1} + \mu_0\theta_0\sum_{k=1}^{\mathcal{C}-i} p(i+k,0)(1-\theta_0)^{k-1},$$
$$1 \le i \le \mathcal{C}-1, \tag{31}$$

$$p(i,j)[\mu_0 + \mu_1 + \lambda\, I_{i+j<\mathcal{C}-1} + \lambda'\, I_{i+j=\mathcal{C}-1}] =$$
$$\frac{p(i,0)\lambda\sigma v\theta_1(1-\sigma)^{j-1}}{\theta_1+\sigma-\sigma\theta_1} + \lambda\sigma v\sum_{k=1}^{j-1} p(i,j-k)(1-\sigma)^{k-1}$$
$$+ \frac{p(0,j)\lambda\sigma u\theta_0(1-\sigma)^{i-1}}{\theta_0+\sigma-\sigma\theta_0} + \lambda\sigma u\sum_{k=1}^{i-1} p(i-k,j)(1-\sigma)^{k-1}$$
$$+ \mu_0\theta_0\sum_{k=1}^{\mathcal{C}-j-i} p(i+k,j)(1-\theta_0)^{k-1} + \mu_1\theta_1\sum_{k=1}^{\mathcal{C}-i-j} p(i,j+k)(1-\theta_1)^{k-1},$$
$$1 \le i \le \mathcal{C}-2,\ \ 1 \le j \le \mathcal{C}-1-i, \tag{32}$$

$$p(\mathcal{C},0)\mu_0 = p(\mathcal{C}-1,0)\lambda' u + \frac{p(0,0)\lambda u\theta_0(1-\sigma)^{\mathcal{C}-1}}{\theta_0+\sigma-\sigma\theta_0} + \lambda u\sum_{k=2}^{\mathcal{C}-1} p(\mathcal{C}-k,0)(1-\sigma)^{k-1}, \tag{33}$$

$$p(0,\mathcal{C})\mu_1 = p(0,\mathcal{C}-1)\lambda' v + \frac{p(0,0)\lambda v\theta_1(1-\sigma)^{\mathcal{C}-1}}{\theta_1+\sigma-\sigma\theta_1} + \lambda v\sum_{k=2}^{\mathcal{C}-1} p(0,\mathcal{C}-k)(1-\sigma)^{k-1}, \tag{34}$$

$$p(i,\mathcal{C}-i)[\mu_0+\mu_1] = \lambda'[p(i-1,\mathcal{C}-i)u + p(i,\mathcal{C}-i-1)v]\, I_{i>1}\, I_{i<\mathcal{C}-1}$$
$$+ \lambda'\frac{p(0,\mathcal{C}-1)u\theta_0}{\theta_0+\sigma-\sigma\theta_0}\, I_{i=1} + \lambda'\frac{p(\mathcal{C}-1,0)v\theta_1}{\theta_1+\sigma-\sigma\theta_1}\, I_{i=\mathcal{C}-1}$$
$$+ \frac{p(i,0)\lambda v\theta_1(1-\sigma)^{\mathcal{C}-i-1}}{\theta_1+\sigma-\sigma\theta_1} + \lambda v\sum_{k=2}^{\mathcal{C}-i-1} p(i,\mathcal{C}-i-k)(1-\sigma)^{k-1}$$
$$+ \frac{p(0,\mathcal{C}-i)\lambda u\theta_0(1-\sigma)^{i-1}}{\theta_0+\sigma-\sigma\theta_0} + \lambda u\sum_{k=2}^{i-1} p(i-k,\mathcal{C}-i)(1-\sigma)^{k-1},$$
$$1 \le i \le \mathcal{C}-1. \tag{35}$$

C Joint queue-length distribution immediately after a departure

Consider a $\mathcal{S}_{2\times 2}(GE/GE/1/\mathcal{C})$ system as depicted in Figure 5. Let (n_0, n_1) denote the state of the system. The solution of the balance equations given in Appendix B yields the steady-state distribution $p(n_0, n_1)$, $0 \le n_0 \le \mathcal{C}$, $0 \le n_1 \le \mathcal{C} - n_0$. Let $r_x(n_0, n_1)$, be the probability that the system is is in state (n_0, n_1) immediately after a departure from queue$-x$, $x = 0, 1$. Then, the distribution $r_x(n_0, n_1)$ can be written in terms of $p(n_0, n_1)$ as follows:

$$
\begin{aligned}
r_0(0, n_1) &= \frac{\mu_0}{H_0} \sum_{i=1}^{\mathcal{C}-n_1} p(i, n_1)(1-\theta_0)^{i-1}, & (36)\\
r_0(n_0, n_1) &= \frac{\mu_0\theta_0}{H_0} \sum_{i=1}^{\mathcal{C}-n_0-n_1} p(i, n_1)(1-\theta_0)^{i-1}, & (37)\\
r_1(n_0, 0) &= \frac{\mu_1}{H_1} \sum_{i=1}^{\mathcal{C}-n_0} p(n_0, i)(1-\theta_1)^{i-1}, & (38)\\
r_1(n_0, n_1) &= \frac{\mu_1\theta_1}{H_1} \sum_{i=1}^{\mathcal{C}-n_0-n_1} p(n_0, i)(1-\theta_1)^{i-1}. & (39)
\end{aligned}
$$

The normalising constants H_0 and H_1 are obtained from the normalising conditions as follows:

$$
\begin{aligned}
H_0 &= \mu_0 \sum_{n_1=0}^{\mathcal{C}-1} \sum_{i=1}^{\mathcal{C}-n_1} p(i, n_1)(1-\theta_0)^{i-1} + \mu_0\theta_0 \sum_{n_0=1}^{\mathcal{C}-1} \sum_{n_1=0}^{\mathcal{C}-1-n_0} \sum_{i=1}^{\mathcal{C}-n_0-n_1} p(i, n_1)(1-\theta_0)^{i-1},\\
H_1 &= \mu_1 \sum_{n_0=0}^{\mathcal{C}-1} \sum_{i=1}^{\mathcal{C}-n_0} p(n_0, i)(1-\theta_1)^{i-1} + \mu_1\theta_1 \sum_{n_0=0}^{\mathcal{C}-1} \sum_{n_1=1}^{\mathcal{C}-1-n_0} \sum_{i=1}^{\mathcal{C}-n_0-n_1} p(n_0, i)(1-\theta_1)^{i-1}.
\end{aligned}
$$

Acknowledgement

The authors would like to thank the referees who took the time to make very detailed and constructive comments on the first draft, resulting in a much better exposition.

References

[1] D. M. Dias and J. R. Jump, "Analysis and Simulation of Buffered Delta Networks," *IEEE Trans. Comput.*, **C-30**, pp. 273–282, Apr. 1981.

[2] L. R. Goke and G. J. Lipovski, "Banyan Networks for Partitioning Multiprocessor Systems," in *Proc. First Annual Symposium on Computer Architecture*, pp. 21–28, 1973.

[3] P. G. Harrison and N. M. Patel, "The Representation of Multistage Interconnection Networks in Queueing Models of Parallel Systems," *Journal of the ACM*, **37**, no 4, pp. 863–898, Oct. 1990.

[4] P. G. Harrison and A. de C. Pinto, "Blocking in Asynchronous, Buffered Banyan Networks," In *Proc. of the IFIP WG 7.3 International Conference on the Performance of Distributed Systems and Integrated Communication Networks*, Kyoto, Japan, ed. T. Hasegawa, H. Takagi and Y. Takahashi, North-Holland, pp. 169–188, 1992.

[5] P. G. Harrison, "On Nonuniform Packet-switched Delta Networks and the Hot-spot Effect," *IEE Proceedings-E*, 138, no 3, pp. 123–130, May 1991.

[6] Y. Jenq, "Performance Analysis of a Packet Switch Based on Single-Buffered Banyan Network," *IEEE Journal on Selected Areas of Communications*, **SAC–1**, no 6, pp. 1014–1021, 1983.

[7] K. P. Jun and H. G. Perros, "An Approximate Analysis of Tandem Queueing Networks with Blocking and General Service Times," *European Journal of Operational Research*, **46**, pp. 123–135, 1990.

[8] H. S. Kim and A. Leon-Garcia, "Performance of Buffered Banyan Networks under Nonuniform Traffic Patterns," *IEEE Trans. Comm.*, **38**, no 5, pp. 648–658, May 1990.

[9] D. Kouvatsos and N. Xenios, "MEM for Arbitrary Queueing Networks with Multiple General Servers and Repetitive-service Blocking," *Performance Evaluation*, **10**, pp. 169–195, Sep. 1989.

[10] C. P. Kruskal and M. Snir, "The Performance of Multistage Interconnection Networks for Multiprocessors," *IEEE Trans. Comput.*, **C–32**, no 12, pp. 1091–1098, Dec. 1983.

[11] C. P. Kruskal, M. Snir and A. Weiss, "The Distribution of Waiting Times in Clocked Multisage Interconnection Network," *IEEE Trans. Comput.*, **37**, no 11, pp. 1337–1352, Nov. 1988.

[12] P. J. Kuehn, "Approximate Analysis of General Queuing Networks by Decomposition," *IEEE Trans. Comm.*, **COM–27**, no 1, pp. 113–126, Jan. 1979.

[13] J. Labarta, J. Domingo and O. Casals, "Performance Evaluation of Packet- switched Omega Networks with Finite Buffers," In *Proc. First Int. Workshop on Queueing Networks with Blocking*, North-Holland, pp. 249–255, 1989.

[14] T. Lin and L. Kleinrock, "Performance Analysis of Finite-Buffered Multistage Interconnection Networks with a General Traffic Pattern," In *Proc. ACM SIGMETRICS '91*, pp. 68–89, 1991.

[15] A. Merchant, "A Markov Chain Approximation for the Analysis of Banyan Networks," In *Proc. ACM SIGMETRICS '91*, pp. 60–67, 1991.

[16] I. Mitrani, *Modelling of Computer and Communication Systems*, Cambridge University Press, 1987.

[17] R. O. Onvural and H. G. Perros, "Approximate Analysis of Multi-Class Tandem Open Queueing Networks with Coxian Parameters and Finite Buffers," In *Proc. Performance'90*, eds. P.J.B. King, I. Mitrani and R.J. Pooley, pp. 131–143, 1990.

[18] J. H. Patel, "Performance of Processor-Memory Interconnections for Multiprocessors," *IEEE Trans. Comput.*, **C-30**, pp. 771–780, Oct. 1981.

[19] N. M. Patel, "Models of Circuit-Switched Interconnection Networks," PhD thesis, Dept. of Computing, Imperial College, University of London, 1989.

[20] A. de C. Pinto, "Models of Finite-buffered Packet-switched Multistage Interconnection Networks," PhD thesis, Dept. of Computing, Imperial College, University of London, in preparation.

[21] T. Szymanski and S. Shaikh, "Markov Chain Analysis of Packet-Switched Banyans with arbitrary Switch Sizes, Queue Sizes, Link Multiplicities and Speedups," *1989 IEEE INFOCOM*, pp. 960–971, 1989.

[22] T. H. Theimer, E. P. Rathgeb and M. N. Huber, "Performance Analysis of Buffered Banyan Networks,"*Performance of Distributed and Parallel Systems*, Kyoto University, pp. 57–72, 1988.

[23] W. Whitt, "The Queueing Network Analyzer," *The Bell System Technical Journal*, **62**, no 9, pp. 2779–2813, Nov. 1983.

[24] D. L. Willick and D. L. Eager, "An Analytical Model of Multistage Interconnection Networks," In *Proc. of ACM SIGMETRICS '90*, pp. 192–199, 1990.

[25] R. W. Wolff, "Poisson Arrivals See Time Averages," *Operations Research*, **30**, pp. 223–231, 1982.

[26] C. Wu and T. Feng, "On a Class of Multistage Interconnection Networks," *IEEE Trans. Comput.*, **C-29**, pp. 694–702, Aug. 1980.

[27] H. Yoon, K. Y. Lee and M. T. Liu, "Performance Analysis of Multibuffered Packet-Switching Networks in Multiprocessor Systems," *IEEE Trans. Comput.*, **39**, no 3, pp. 319–327, Mar. 1990.