An Analytical Performance Model for Multistage Interconnection Networks with finite, infinite and zero length buffers [⋆,⋆⋆]

C. Bouras, J. Garofalakis, P. Spirakis, V. Triantafillou

Department of Computer Engineering and Informatics,
Patras University, Rio, 265 00 Patras, Greece, and
Computer Technology Institute,
P.O. Box 1122, 261 10 Patras, Greece
E-mail : {bouras, garofala, spirakis, triantaf}@cti.gr

Abstract

Multistage Interconnection Networks (MINs) with crossbar switches have been used to interconnect processors and memory modules in parallel multiprocessor systems. They also play an increasingly important role in the development of ATM networks. In this paper we analyze the general case of MINs, made of $k \times k$ switches with finite, infinite or zero length buffers (unbuffered). The exact solution of the steady state distribution of the first stage is derived for all cases. We use this to get an approximation for the steady state distributions in the second stage and beyond. In the case of unbuffered switches we reach the known exact solution for all the stages of the MIN. Our results are validated by extensive simulations.

Key words: analytical models, queueing theory models, evaluation.

1 Introduction

Multistage Interconnection Networks (MINs) have attracted from the early '80s the attention of the designers of highly parallel multiprocessor systems

⋆ This research was partially supported by the European Union ESPRIT Basic Research Projects ALCOM IT (contract no. 20244) and GEPPCOM (contract no. 9072) and the Greek Ministry of Education.
⋆⋆A short version of this paper ([4]) has appeared in Euro–Par '97

with a large number of processors. They are required to provide high bandwidth to support the communication between processors and memory modules. MINs (which are packet-switched) have been adopted in the past in several machines ([3],[13]) and are expected also to play an important role in the development of high-speed networks based on Asynchronous Transfer Mode (ATM) [15]. The performance of a MIN is of crucial importance, thus a lot of research has been dedicated to the study of how these networks perform under various conditions, through analytic techniques or simulation ([1], [2], [6], [7], [8], [9], [11], [12], [14]). Analytic results can be found for specific cases of MINs, which mainly rely on approximation methods.

Koch in [7] proved that an increase of the link-bandwidth of unbuffered butterfly networks by a constant factor increases their throughput by more than a constant factor. Bouras et.al. [1] provided nearly tight upper bounds on the mean delays of the second stage and beyond, in the case of infinite buffers and validated their results by simulations. Their analysis indicated that after the second stage there is no notable difference between the delay times, giving a partial positive answer to the conjecture and experimental results of [8]. Merchant [11] approximated the underlying non-marcovian processes by marcov models. His marcov approximation of the throughput of finite and infinite buffered MINs under uniform and non-uniform traffic is validated by comparisons to simulation results. Garofalakis and Spirakis [6] analyzed Banyan networks with finite buffers, providing the exact solution of the steady-state distribution of the first stage in the situation where packets are lost when they encounter a full buffer. Rehrmann et.al. [14] presented an analysis of the commmunication throughput of single-buffered multistage interconnection networks consisting of 2×2 switches and with maximum injection rate $p = 1$, using the *relaxed blocking model* where messages that cannot be routed, due to the fact that a receiving buffer is occupied, are deleted. They show that the throughput is $\Theta(n/\sqrt{logn})$ if n is the size of the network. They also analysed the equilibrium-situation of the network and gived tight upper and lower bounds on the steady-state distribution of I/O sequences.

The basic building block of the packet-switched MINs considered here, is a k-input, k-output ($k \times k$) switch grouped in stages. We examine MINs that provide a unique path from each source (processor) to each sink (memory module), which belong to the class of Banyan MINs [5]. Our work considers *general MINs*, that is, MINs made by switches with finite, infinite or zero length buffers (unbuffered), arbitrary switch size ($k \times k$) and variable injection rate p at the sources. Assuming that the traffic (requests for memory modules) feeding the first stage of the MIN is uniform, that at each cycle a packet is generated with fixed probability p, and that packets are lost when they are attempted to be queued at a full buffer (relaxed blocking model), we derive for the general MIN, the exact steady-state distribution of queue lengths in the first stage, and of course exact formulas for the expected number of packets

lost per cycle, and the mean queue length. We then use the results for the first stage and an operational approximation hypothesis to get the (approximate) distributions of the queue sizes of the second stage and beyond. Extensive simulations verify our results, as we discuss in Section 6.

The assumption of uniform and independent Bernoulli traffic feeding the first stage of the MIN, is of course a simplification of the real world situation, especially regarding the use of MINs in ATM switches. However, this assumption is important in order to reach the exact (first stage) and nearly exact (subsequent stages) analytic solution. This analytic solution may provide insight to the behavior of the MIN in any case (e.g. for the performance of a stage compared to the respective performance of preceding stages), and is expected to give a very good approximation to cases of heavy traffic ($p \approx 1$), which are the most interesting.

Our analysis, based on the theory of recurrence equations, explicitly provides the form of the queue length distribution, which is a linear mixture of geometrics. In the next section we present the model that we use and discuss the equilibrium and interstage dependencies which are the factors that crucialy affect any analytic approach. Sections 3 and 4 provide the basic exact analytic results for the first stage, and Section 5 describes the approximation for the subsequent stages and presents the overal network performance measures. Finally, in Section 6 we compare our results with simulation experiments and discuss the performance of the general MIN.

2 Our Approach

2.1 The Model

MINs are packet switched and they are required to provide high bandwidth to support the communication between processors and memory modules. We consider that the network is built by switches connected by unidirectional lines. General MINs consist of a number of $k \times k$ switches (nodes) grouped into stages (Figures 1 and 2). A k-input, k-output switch, can receive packets at each of its k input ports and send them through each of its k output ports (Figure 2). In each output port there is a buffer. We assume that the buffers may be of infinite, finite or zero length (unbuffered switches). Such a network can be modeled as a labelled digraph where nodes are of the following three types: source nodes (indegree 0, outdegree 1), sink nodes (indegree 1, outdegree 0) or switches (positive indegree and outdegree). In this labelled digraph each edge represents one or more lines going from a node to its succesor.

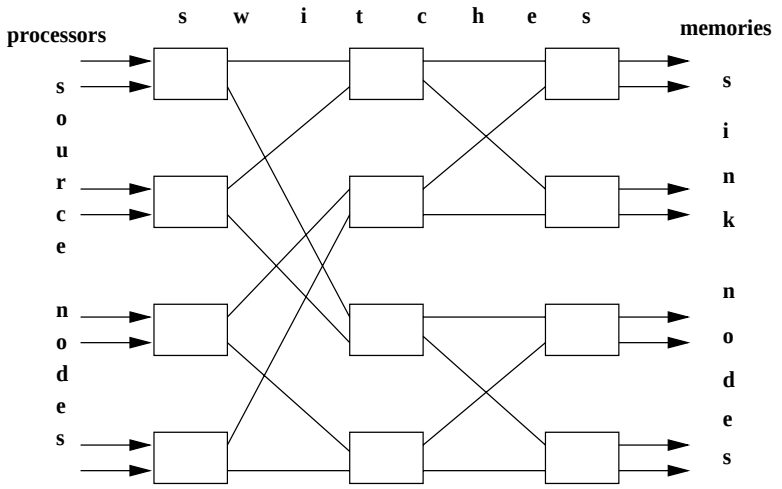

Fig. 1. MIN with 3 stages and 2×2 switches.

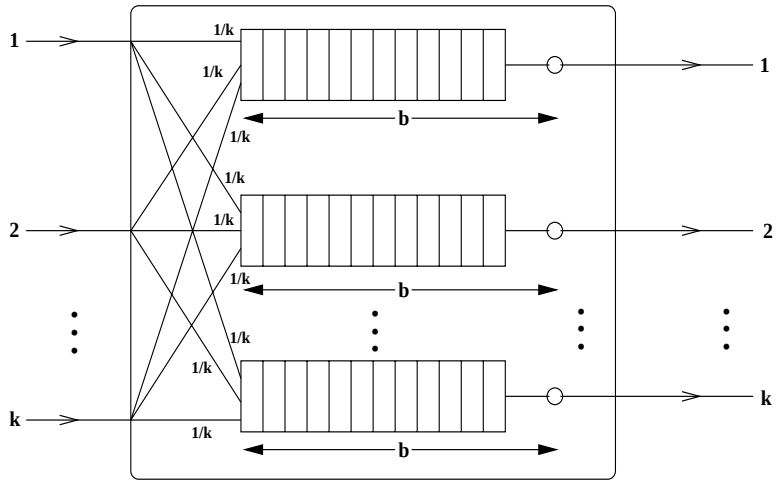

Fig. 2. A k-input , k-output ($k \times k$) switch with buffer size $b \geq 1$.

If there is a unique path from each processor (source node) to each memory module (sink node) then a MIN belongs to the class of Banyan Networks (BNs). We assume oblivious routing algorithms, i.e. algorithms in which the path of a packet through the network is fixed at the source node issuing it. The path can be encoded as a sequence of labels of the successive switch outputs of the path (path descriptor). Packets are generated at each processor by independent, identically distributed random processes. In our analysis we assume that each processor generates a packet with probability p at each cycle, and sends this with equal probability to any memory module (uniform access). The switches have a FIFO policy for their servers (outputs). Conflicts between packets simultaneously routed to the same output port are resolved by queueing the packet. Our analysis assumes that packets are lost when they are attempted to be queued at a full queue or in the case with unbuffered switches. In actual parallel machines, the sending processor is notified, in order to resubmit the packet later on. The service time of the output queues of each switch is assumed constant and equal to the network cycle time. The

uniform access assumption allows us to represent any $k \times k$ switch as a system of k queues working in parallel, with a deterministic server each (of service time equal to 1). Any packet which enters any of the k inputs of the switch, goes with probability $1/k$ to any of the (output) queues of the switch. In our analysis we assume that the buffer length b includes the server (output). So, an unbuffered switch is referred with $b = 1$. We assume that arrivals happen at the end of each cycle (thus first the queue is served and then new packets arrive, if any). The routing logic at each switch is assumed to be fair, i.e. conflicts are randomly resolved.

2.2 The Equilibrium and Interstage Dependencies

Most authors that have used analytic approaches for the analysis of MIN's, have remarked the basic difficulty for any analytic approach. Except for the case of unbuffered switches ([8], [12]) in all other cases, the traffic flow between consecutive stages depends upon time, that is the distribution of packet arrivals at the second and the subsequent stages is not time independent, as is the case for the first stage which is feeded by the independent "Bernoulli" processors. ([8], [10], [11], [14]). However, in [14] it is pointed out that the behaviour, say b_t, of a stage at time t depends mainly upon the present, a little bit ($b_{t-1}/4$) upon the situation at time $t-1$, and is nearly independent ($b_{t-r}/4^r$) from ancient events at time $t-r$. So, the dependency from history is exponentially desreasing. This last observation, together with the assumption that every stage of the MIN will reach an equilibrium (steady-state), leads to the marcovian approximation which we present in section 5. *The output queues of stage m that feed the stage $m+1$, are assumed to operate like independent "Bernoulli" processors with a packet generation probability equal to their utilization.* Clearly, this hypothesis equates the dynamics of the output process of a stage with its "macroscopic" averages, ignoring any time dependency of its behaviour.

The assumption that the stages of a MIN reach a steady-state, is validated not only by our simulation experiments, but also by the fact that it is true for the first stage (which actually feeds the subsequent stages of the network), where our approach presents the exact solution for any MIN.

Futhermore, our markovian approximation provides a unified framework for the performance evaluation of MINs, which when it is applied to the case of unbuffered swithes, leads to the known exact solution, expressed by a recurrence equation ([8], [12]). This gives strong evidence that our approximation provides nearly exact solutions, which tend to the exact solution, as the buffer size is decreased down to the unbuffered case, which we treat simply as a case of the general buffered switched MINs (with buffer size equal to 1). As

we discuss later in this paper, our approximation agrees extremely well to the simulation experiments that we have performed, not only for the mean steady-state metrics, but also for the distribution of packets in the switches, as well.

3 The General Recurrence Relation for the First Stage

Let C be the random variable denoting the number of packets arriving to an output buffer of an $k \times k$ switch of the first stage of the network at the end of a cycle and

$$x_{k,c} = \mathbf{Pr}(C = c)$$

Some of these arriving packets may be lost due to a full queue.

Lemma 1 *The arrival process of packets at the output queues of the first stage of the network, is given by a Binomial distribution bin$(k, p/k)$, where p is the fixed probability of a packet generated by a processor at each cycle. Therefore we have*

$$x_{k,c} = \begin{cases} \binom{k}{c} (\frac{p}{k})^c (1 - \frac{p}{k})^{k-c}, & for\ 0 \leq c \leq k \\ 0, & otherwise \end{cases} \quad (1)$$

Definition 2 *Let $q^{(n)}$ be the number of packets in an arbitrary output queue at the end of the cycle n and let q be the steady state limit of $q^{(n)}$.*

Definition 3 *Let $\upsilon^{(n)}$ be the number of packets that are entering an arbitrary output queue at cycle n and let υ be the steady state limit of $\upsilon^{(n)}$. It holds that $\upsilon^{(n)} \leq C$ at each cycle n, when b is finite. If b is infinite, it is always true that $\upsilon^{(n)} = C$.*

Definition 4 *Let $p_j = \mathbf{Pr}(q = j)$, $j \geq 0$, be the distribution of q at the steady state. Also, $p_{0,1} = p_0 + p_1$*

Lemma 5 *For $0 < m \leq min(b, k)$:*

$$\mathbf{Pr}(\upsilon^{(n)} = m) = \begin{cases} x_{k,m} & if q^{(n-1)} - \Delta(q^{(n-1)}) < b - m, \\ (x_{k,m} + x_{k,m+1} + \cdots + x_{k,k}) & if q^{(n-1)} - \Delta(q^{(n-1)}) = b - m, \\ 0 & otherwise \end{cases} \quad (2)$$

where $\Delta(q^{(n)})$ is the departure of a packet from an arbitrary output queue at the end of cycle n, if any.

For $m = 0$, $\mathbf{Pr}(v^{(n)} = 0) = x_{k,0}$ for any $q^{(n-1)}$.

Obviously, for $m > \min(b, k)$, $\mathbf{Pr}(v^{(n)} = m) = 0$.

PROOF. If $q^{(n-1)}$ is the number of packets in the buffer during cycle $n-1$, at the end of the cycle after some possible arrivals, there will be left $\mathcal{F}^{(n-1)} = q^{(n-1)} - \Delta(q^{(n-1)})$ packets in the buffer ($\Delta(q^{(n-1)})$ is the departure of a packet, if there is one in the buffer). Thus, if $\mathcal{F}^{(n-1)} + m < b$, the buffer will not become full if there will arrive up to m packets at the end of the cycle $n-1$ and also $\mathbf{Pr}(v^{(n)} = m) = x_{k,m}$. If $\mathcal{F}^{(n-1)} + m > b$, obviously the buffer cannot receive m packets and $\mathbf{Pr}(v^{(n)} = m) = 0$. If $\mathcal{F}^{(n-1)} + m = b$, after m packets entering the buffer, it will be full, thus the extra packets (if any) will be lost. So, $\mathbf{Pr}(v^{(n)} = m) = \mathbf{Pr}$(either m arrive, either $m+1$ arrive, ..., either k arrive) = $= (x_{k,m} + x_{k,m+1} + \cdots + x_{k,k})$. It always holds that $\mathbf{Pr}(v^{(n)} = 0) = x_{k,0}$, since at the end of the cycle $n-1$, a packet will leave from the buffer in any case (if there is one). Clearly, when we have infinite size buffers ($b = \infty$), we have $\mathbf{Pr}(v^{(n)} = m) = x_{k,m}$, always.

Theorem 6 *The steady state flow balance equations are : ($p_{0,1} = p_0 + p_1$)*

$$\begin{aligned} p_0 &= p_{0,1} x_{k,0} \\ p_1 &= p_{0,1} x_{k,1} + p_2 x_{k,0} \\ p_2 &= p_{0,1} x_{k,2} + p_2 x_{k,1} + p_3 x_{k,0} \\ &\vdots \\ p_k &= p_{0,1} x_{k,k} + p_2 x_{k,k-1} + \cdots + p_{k+1} x_{k,0} \end{aligned} \tag{3}$$

while for $k \le j < b$, the general recurence holds :

$$\begin{aligned} & p_j x_{k,0} = p_{j-k+1}(x_{k,k}) + p_{j-k+2}(x_{k,k-1} + x_{k,k}) + \cdots + \\ & p_{j-2}(x_{k,3} + x_{k,4} + \cdots + x_{k,k}) + \\ & p_{j-1}(x_{k,2} + x_{k,3} + \cdots + x_{k,k}), \quad k \le j < b \end{aligned} \tag{4}$$

The same equation (4) holds for $j = b$, in the case of finite buffers, or unbuffered switches ($b = 1$).

PROOF. Starting with the case $b = \infty$, we have

$$\mathbf{Pr}(q^{(n)} = 0) = \mathbf{Pr}(q^{(n-1)} = 0 \text{ and } v^{(n)} = 0) + \mathbf{Pr}(q^{(n-1)} = 1 \text{ and } v^{(n)} = 0)$$

This, in steady state, becomes

$$\mathbf{Pr}(q = 0) = \mathbf{Pr}(q = 0 \text{ and } v = 0) + \mathbf{Pr}(q = 1 \text{ and } v = 0)$$

Since arrivals are independent of the queue size when there is enough room in the buffers, we get

$$\begin{aligned} p_0 &= p_0 x_{k,0} + p_1 x_{k,0}, \text{ or} \\ p_0 &= p_{0,1} x_{k,0} \end{aligned}$$

In the same way we get

$$\begin{aligned} p_1 &= p_0 x_{k,1} + p_1 x_{k,1} + p_2 x_{k,0} \\ p_2 &= p_0 x_{k,2} + p_1 x_{k,2} + p_2 x_{k,1} + p_3 x_{k,0} \\ &\vdots \\ p_k &= p_0 x_{k,k} + p_1 x_{k,k} + p_2 x_{k,k-1} + \cdots + p_k x_{k,1} + p_{k+1} x_{k,0} \end{aligned}$$

For $k < n < b$ we get

$$p_n = p_{n-k+1} x_{k,k} + p_{n-k+2} x_{k,k-1} + \cdots + p_n x_{k,1} + p_{n+1} x_{k,0}$$

By setting $p_{0,1} = p_0 + p_1$ we now have

$$\begin{aligned} p_0 &= p_{0,1} x_{k,0} \\ p_1 &= p_{0,1} x_{k,1} + p_2 x_{k,0} \\ &\vdots \\ p_k &= p_{0,1} x_{k,k} + p_2 x_{k,k-1} + \cdots + p_k x_{k,1} + p_{k+1} x_{k,0} \end{aligned} \tag{5}$$

and

$$p_n = p_{n-k+1} x_{k,k} + p_{n-k+2} x_{k,k-1} + \cdots + p_n x_{k,1} + p_{n+1} x_{k,0} \tag{6}$$

for $k < n < b$.
By adding $p_0 + \cdots + p_{k-1}$ and using $x_{k,0} + x_{k,1} + \cdots + x_{k,k} = 1$ we get

$$p_k x_{k,0} = p_{0,1} x_{k,k} + p_2 (x_{k,k-1} + x_{k,k}) + \cdots + p_{k-1} (x_{k,2} + \cdots + x_{k,k}) \tag{7}$$

By using (6) for $n = k + 1$, with (7), and working inductively, we finally get

$$\begin{aligned} p_n x_{k,0} = p_{n-k+1} x_{k,k} + \cdots + p_{n-2}(x_{k,3} + \cdots + x_{k,k}) + \cdots + \\ p_{n-1}(x_{k,2} + \cdots + x_{k,k}) \end{aligned} \tag{8}$$

for $k \leq n < b$
Equation (8) proves Theorem 6 for $b = \infty$.

For $b < \infty$, the situation for p_b, the steady-state probability of having a full buffer, has to be given special consideration:

$$\begin{aligned} \mathbf{Pr}(\mathbf{q^{(n)} = b}) = & \mathbf{Pr}(q^{(n-1)} = b \text{ and } v^{(n)} = 1) \\ & + \mathbf{Pr}(q^{(n-1)} = b - 1 \text{ and } v^{(n)} = 2) \\ & \vdots \\ & + \mathbf{Pr}(q^{(n-1)} = b - k + 1 \text{ and } v^{(n)} = k) \end{aligned}$$

But
$\mathbf{Pr}(q^{(n-1)} = b - k + \lambda \text{ and } v^{(n)} = k - \lambda + 1) =$
$\mathbf{Pr}(q^{(n-1)} = b - k + \lambda)\mathbf{Pr}(v^{(n)} = k - \lambda + 1 \text{ given that } q^{(n-1)} = b - k + \lambda) =$
$\mathbf{Pr}(q^{(n-1)} = b - k + \lambda)(x_{k,(k-\lambda)+1} + x_{k,(k-\lambda)+2} + \cdots + x_{k,k})$

In steady state we get

$$\begin{aligned} p_b \quad &= p_b(x_{k,1} + x_{k,2} + \cdots + x_{k,k}) + p_{b-1}(x_{k,2} + \cdots + x_{k,k}) + \cdots + p_{b-k+1} x_{k,k} \\ &or \\ p_b x_{k,0} &= p_{b-k+1} x_{k,k} + p_{b-k+2}(x_{k,k-1} + x_{k,k}) + \cdots + p_{b-1}(x_{k,2} + \cdots + x_{k,k}) \end{aligned}$$

By the same way, working inductively, one may show that for $k \leq j < b$ we get :

$$p_j x_{k,0} = p_{j-k+1} x_{k,k} + p_{j-k+2}(x_{k,k+1} + x_{k,k}) + \cdots + p_{j-1}(x_{k,2} + \cdots + x_{k,k}),$$

thus proving Theorem 6 , for $b < \infty$ (including the unbuffered case $b = 1$).

4 Solution of the First Stage

The characteristic equation for the above recurence relation (4) is,
for $b \geq j \geq k$ ($b < \infty$ or $b = \infty$) :

$$F(y) = 0, \tag{9}$$

where

$$F(y) = x_{k,0}y^{k-1} - (x_{k,2} + x_{k,3} + \ldots + x_{k,k})y^{k-2} - \ldots - (x_{k,k-1} + x_{k,k})y - x_{k,k}$$

CASE 1: $F(y)$ has distinct roots $R_1, \ldots, R_{k-1}$. Then the steady-state probabilities are

$$p_j = A_1R_1^{j-1} + A_2R_2^{j-1} + \cdots + A_{k-1}R_{k-1}^{j-1} \tag{10}$$

where $A_1, A_2, \ldots, A_{k-1}$ are constants that can be derived from the initial conditions

$$\begin{aligned} p_{0,1} &= A_1 + A_2 + \cdots + A_{k-1} \\ p_2 &= A_1R_1 + A_2R_2 + \cdots + A_{k-1}R_{k-1} \\ &\vdots \\ p_{k-1} &= A_1R_1^{k-2} + A_2R_2^{k-2} + \cdots + A_{k-1}R_{k-1}^{k-2} \end{aligned} \tag{11}$$

together with $p_{0,1} = p_0 + p_1$, $\sum_{n=0}^{b} p_n = 1$, ($b < \infty$ or $b = \infty$) and the equations (3) for p_0, p_1, ..., p_{k-2}.

CASE 2: $F(y)$ has at least one multiple nonzero root. Then the system is unstable, that is

$$\lim_{n\to\infty} q^{(n)} = \infty$$

Theorem 7 (stability criterion) *A steady state queue size distribution exists if and only if $F(y)$ has distinct roots.*

The cases of instability should occur only when $b = \infty$ (infinite buffers) and $p = 1$. However, applying our method for networks with switches 2×2, 3×3 and 4×4, we never faced the above CASE 2. It is an open problem to prove these observations analytically.

It is easy to derive analytically the roots $R, \ldots R_{k-1}$ when $k = 2, 3, 4, 5$ that is for networks with 2×2, 3×3, 4×4, 5×5 switches, which are, anyway, the most interesting cases. For switches with $k > 5$, we must rely on arithmetic methods in order to solve the $(k-1)$-degree recurence relation.

As we remark in Section 6, the 2×2 switches perform better compared to switches with higher k, as far as the mean number of lost packets per cycle are concerned. So, we present here analytic results for 2×2 switches except for the case of unbuffered switches, where we can easily derive results for the general case $k \times k$. In Section 6, we present numerical results for 3×3 switches also, derivered analytically.

4.1 Switches with finite buffers ($b < \infty$)

By using (5) for $k = 2$, we get one root R_1, which, given that $x_{2,0} = (1-p/2)^2$ and $x_{2,2} = p^2/4$, is:

$$R_1 = \frac{x_{2,2}}{x_{2,0}} = (\frac{p}{2-p})^2 \tag{12}$$

The constant A_1 is given by

$$A_1 = \begin{cases} \frac{1-R_1}{1-R_1^b}, & \text{for } p < 1 \\ \frac{1}{b}, & \text{for } p > 1 \end{cases} \tag{13}$$

The steady state probabilities are :

$$\left.\begin{aligned} p_0 &= A_1 x_{2,0} \\ p_1 &= A_1(1 - x_{2,0}) \\ p_j &= A_1 R_1^{j-1}, \qquad 2 \le j \le b \end{aligned}\right\} \text{ for } p < 1 \tag{14}$$

or

$$\left.\begin{aligned} p_0 &= 1/4b \\ p_1 &= 3/4b \\ p_j &= 1/b, \;\; 2 \le j \le b \end{aligned}\right\} \text{ for } p = 1 \tag{15}$$

By using the above steady-state probabilities of (14) or (15), we derive the *mean number of packets in an output queue of the first stage*:

$$E(q) = \sum_{j=0}^{b} j p_j = p + \frac{p^2[1 - p_b(1 - p + b)]}{4(1-p)}, \;\; \text{for } p < 1 \text{ and } b > 1 \tag{16}$$

and

$$E(q) = \sum_{j=0}^{b} jp_j = \frac{b+1}{2} - \frac{1}{4b}, \text{ for } p = 1 \text{ and } b > 1$$

It is worth pointing out that for $b \to \infty$, we get $p_b \to 0$ much faster, thus equation (16) agrees with the known formula of [8] for the infinite buffer case (equation 23).

For the *mean number of packets lost in a cycle at an output queue of the first stage* we have:

for $p < 1$: Mean number of arriving packets $E(C) = p$ and

$$E(v) = \begin{cases} p - (p^2/4)p_b, \, b > 1 \\ p - (p^2/4), \quad b = 1 (\text{unbuffered switch}) \end{cases} \tag{17}$$

thus :

$$E(\text{packets lost in one cycle}) = E(C) - E(v) = \begin{cases} (p^2/4)p_b, \, b > 1 \\ p^2/4, \qquad b = 1 \end{cases} \tag{18}$$

for $p = 1$:

$$E(C) = 1, E(v) = 1 - \frac{1}{4b} \tag{19}$$

$$E(\text{packets lost in one cycle}) = E(C) - E(v) = \begin{cases} 1/4b, \, b > 1 \\ 1/4, \quad b = 1 \end{cases} \tag{20}$$

4.2 Switches with infinite buffers ($b = \infty$)

In this case we have $b = \infty$, thus for $k = 2$, we get the same

$$x_{2,0} = (1 - p/2)^2, x_{2,2} = p^2/4$$

and the root $R_1 = (\frac{p}{2-p})^2$. The difference is in the constant A_1 which is now :

$$A_1 = 1 - R_1 \tag{21}$$

The steady-state probabilities are :

$$\left.\begin{aligned} p_0 &= 1-p \\ p_1 &= A_1(1-x_{2,0}) \\ p_j &= A_1 R_1^{j-1}, \qquad j \geq 2 \end{aligned}\right\} \text{ for } p<1 \tag{22}$$

For $p=1$ we don't have steady-state probabilities, since this is an instability case. Equations (22) are in agreement with [8], since they provide the known result:

$$E(q) = p + \frac{p^2}{4(1-p)} \tag{23}$$

4.3 Unbuffered switches ($b=1$)

For the general case ($k \times k$ switches), we have two balance equations :

$$\begin{aligned} p_0 &= p_0 x_{k,0} + p_1 x_{k,0} = x_{k,0} \\ p_1 &= p_0(x_{k,1} + x_{k,2} + \cdots x_{k,k}) + p_1(x_{k,1} + x_{k,2} + \cdots x_{k,k}) \\ &= 1 - x_{k,0} \end{aligned} \tag{24}$$

Since $x_{k,0} = (1-p/k)^k$, we have

$$p_1 = 1 - (1-p/k)^k \tag{25}$$

Equation (25) is exactly the equation $P_{m+1} = 1-(1-P_m/k)^k$ of [12] and [8], when $m=0$. We may remark here, that the above authors, derive this equation for all the stages of the network. This is an evidence that our approximation for the stages beyond the first stage (section 5) is valid even for the cases when $b>1$. Easily, we get

$$\begin{aligned} E(q) &= 1 - x_{k,0} = 1 - (1-p/k)^k \\ E(lost) &= p - 1 + x_{k,0} = p - 1 + (1-p/k)^k \end{aligned} \tag{26}$$

The last equation is the same with (18) for $b=1$, when $k=2$, as expected.

5 Subsequent Stages and Network Performance

In accordance to the remarks stated in Section 2.2, we assume now the following approximation hypothesis :

Hypothesis : The output queues of stage m that feed stage $m+1$, are assumed to operate like processors with a packet generation probability $p_{(m)}$ such that

$$p_{(m)} = \text{utilization of an output queue of stage } m \text{ (and } p_{(0)} = p)$$

This hypothesis equates the dynamics of the output process of a stage with its "macroscopic" averages.

Definition 8 *Let $p_{j,i}$ = the steady state probability of finding j packets in an output queue of stage i of the network.*

Suppose that we have a network with L stages. Our approximation scheme is iterative and is described in Algorithm I (Figure 3).

```
Algorithm I
    p(0) := p
    FOR i = 1 TO L DO
    BEGIN
        Set p := p(i-1)
        Calculate x_{k,0}, x_{k,1}, ..., x_{k,k},
        Evaluate p_{0,i}, p_{1,i}, ..., p_{b,i}, from equations (10),(11)
        Evaluate E(q), E(lost) for stage i
        p(i) := 1 - p_{0,i}
    END
    CALCULATE NETWORK PERFORMANCE MEASURES
    (BANDWIDTH, AVERAGE TRANSIT TIME etc.)
```

Fig. 3. The Algorithm I

This approximation scheme has the following nice properties :

- It provides an *exact* solution for all stages for unbuffered networks as we commented in Section 4.3
- It approximates not only the average maesures such as $E(q)$ and $E(lost)$, but also the distribution itself of the queue sizes, with a maximum relative error in all cases, less than 5%. Higher errors are observed only in cases where the absolute values are very small and the simulation experiments count only a few respective events (e.g. lost packets when p is small).

6 Comparison with Simulation Results and Discussion

We performed extensive simulations to validate our results. Some indicative results are presented in Figures 4-7. The simulations verify our analysis for the first stage and the subsequent ones for all different cases (unbuffered, infinite and finite buffers). Moreover, they prove that the hypothesis introduced has a strong physical sence.

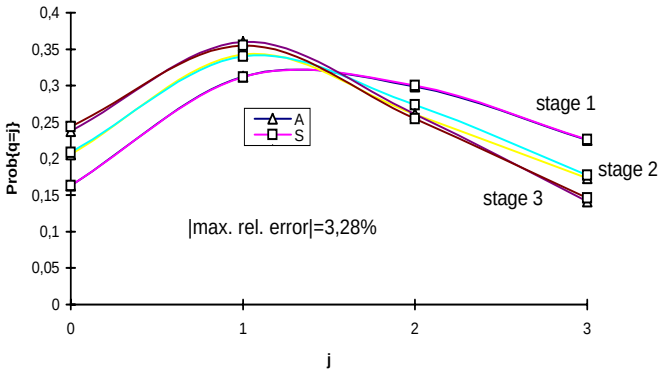

Fig. 4. The distribution of the number q of packets in an output queue of a switch ($3x3$ switches, $p = 0.9$, $b = 3$).

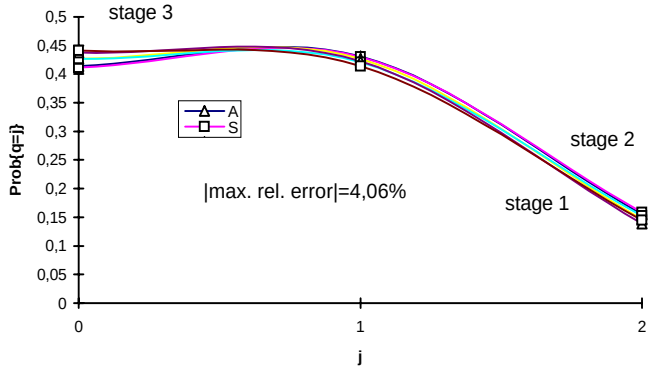

Fig. 5. The distribution of the number q of packets in an output queue of a switch ($2x2$ switches, $p = 0.6$, $b = 2$).

The comparison of the analytic results with the simulation experiments, confirms the exact solution of the first stage for all classes of MINs studied and the fact that the algorithm for the next stages presents an exact solution for all stages in the case of unbuffered network ($b = 1$). Our approximation predicts cumulative performance measures (such as mean queue length) with very small relative error. As far as the steady state distribution of queue sizes is concerned, we approximate the largest steady state probabilities with a very good accuracy, in all stages. For the low-valued probabilities (p_b, p_{b-1}) we odserve a small absolute error and a greater relative one. This error is caused probably due to the fact that the blocking phenomena that relate to these probabilities happen rarely, thus they are encountered a few times by the simulation of the

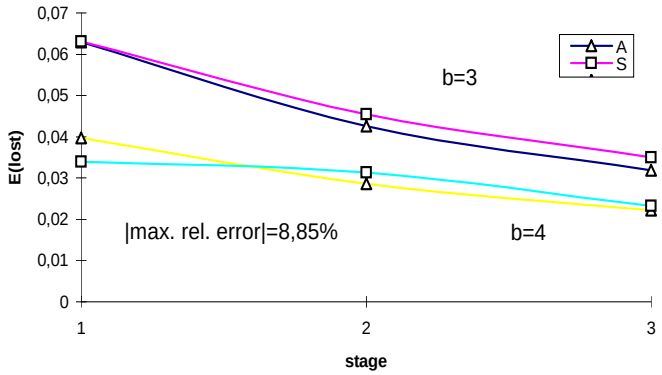

Fig. 6. Mean number of lost packets per cycle at an output queue of a switch ($3x3$ switches, $p = 0.9$).

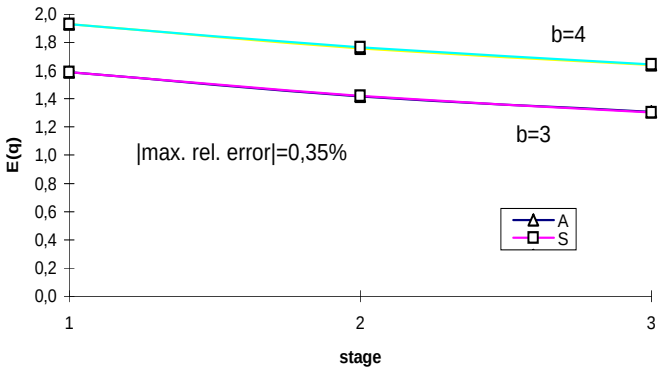

Fig. 7. Mean queue length of an output queue of a switch ($3x3$ switches, $p = 0.9$).

network. A relative error of 5% for the above probabilities, will cause a relative error of about 10% for the mean number of lost packets per queue, for the stages beyond the first since it depends mainly on those small probabilities.

7 Conclusions and future work

Under our analysis, networks with 2×2 switches seem to perform better than the 3×3 switches, with respect to the mean number of packets lost per queue.

- For networks with 2×2 switches :
  For low traffic ($p \leq 0.4$) buffers of size 3 are sufficient to allow only a small fraction (of about 0.0001) of the packets to be lost per queue. The buffer size becomes $b = 8$ for moderate to heave traffic ($0.4 < p \leq 0.8$) and $b = 15$ for very heavy traffic ($0.8 < p \leq 0.9$), respectively in order to keep the losses at the same low level.
- For networks with 3×3 switches :
  The buffers should be respectively of length $b = 4, b = 10, b = 18$ in order to get the same proportion of lost packets per cycle.

We expect that this tendancy - as k increases the mean number of packets lost increases - also holds for networks with greater k. The small fraction of lost packets implies that resubmission of those packets from the processors will not increase the input traffic noticeably. Thus, one can use our analysis to predict the performance of actual networks where lost packets are resubmitted later.

An interesting open problem is to use our analysis in order to obtain analytical results when MINs are adopted in an ATM switching facric. Our approach will be adopted to analyze such networks under different traffic considerations (eg. constant service rates) than those presented in this paper.

Acknowledgements

We acknowledge all the reviewers for their valuable comments and suggestions.

References

[1] C. Bouras, J. Garofalakis, P. Spirakis and V. Triantafillou, *Queueing Delays in Buffered Multistage Interconnection Networks*, Proc. of the 1987 ACM Sigmetrics Conference, May 11–14 1987, Banff, Alberta, Canada, pp. 111–121

[2] D.M. Dias, Jump J.R. *Analysis and simulation of buffered Delta Networks*, IEEE Trans. Computer, vol. C–30, April 1981, pp. 273–282

[3] A. Gottlieb, R. Grishman, C. P. Kruscal, K. P. McAuliffe, L. Rudolph, M. Snir, *The NYU Ultracomputer–Designing an MIMD Shared Memory Parallel Computer*, IEEE Trans. Computers, Vol. C–32, No. 2, Febr. 1983, pp. 175–189

[4] C. Bouras, J. Garofalakis, P. Spirakis, V. Triantafillou, *A General Performance Model for Multistage Interconnection Networks*, Euro–Par '97, Aug. 25–29

[5] G.F. Goke, G.J.Lipovski *Banyan Networks for Partitioning Multiprocessor Systems*, Proc. 1st Ann. Symp. on Computer Architecture, 1973, pp. 21-28

[6] J. Garofalakis, P. Spirakis *The performance of Mutlistage Interconnection Networks with Finite Buffers*, Proc. of the ACM SIGMETRICS Conference, 1990, short paper.

[7] R.R. Koch *Increasing the size of a Network by a constant factor Can Increase Performance by More Than a Constant Factor*, IEEE Symp. on Found. of Comp. Sc. (FOCS 88), pp. 221-231

[8] C.P. Kruskal, M. Snir *The performance of multistage interconnection networks for multiprocessors*, IEEE Trans. Comp., vol. C-32, Dec 1983, pp. 1091-1098

[9] C.P. Kruskal, M. Snir, A. Weiss *The Distribution of Waiting Times in Closed Multistage Interconnection Networks*, IEEE Trans. on Computers, vol. 32, 1988, p. 1337-1352

[10] F.T. Leihton, *Inroduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, Morgan Kaufmann Publishers, 1992

[11] A. Merchant, *A Markov chain approximation for the analysis of Banyan networks*, Proc. ACM Sigmetrics Conf. on Measurement and Modelling of Computer Systems, 1991

[12] J.H.Patel, *Performance of processor–memory interconnection for multiprocessors* IEEE Trans. on Computing, vol. C-30, 1981, pp. 771-780

[13] G. Pfister, M. C. Brantley, D. A. George, S. L. Harvey, W. J. Kleinfelder, K. P. McAuliffe, E. A. Melton, V. A. Norton, J. Weiss, *The IBM Research Parallel Processor Prototype (RP3): Introduction and Architecture*, Proc. 1985 Int. Conf. Parallel Processing, pp. 764-771

[14] R. Rehrmann, B. Monien, R. Luling, R. Diemann, *On the Communication Throughput of Buffered Multistage Interconnection Networks*, ACM SPAA'96, pp. 152-161

[15] R. Rooholaminiight *Finding the right ATM switch for the Market*, IEEE J. Comp., 1994, pp. 16-28