An Approximate Analytical Performance Model for Multistage Interconnection Networks with Backpressure Blocking Mechanism

John Garofalakis
Department of Computer Engineering and Informatics, University of Patras and,
Research Academic Computer Technology Institute Patras, Greece
Email: garofala@cti.gr

Eleftherios Stergiou
Department of Information Technology and Telecommunications, ATEI of Epirus, Greece
Email: ster@teiep.gr

***Abstract*— Multistage Interconnection Networks (MINs) are used to interconnect different processing modules in various parallel systems or on high bandwidth networks. In this paper an integrated performance methodology is presented. A new approximate performance model for self-routing MINs consisting of symmetrical switches which are subject to a backpressure blocking mechanism is analyzed. Based on this, the steady-state distribution of the queue utilization is estimated and then all important performance metrics are calculated. Moreover, a general evaluation factor which helps in choosing a better performance MIN in comparison with other similar MIN architecture specifications is defined. The model was exemplified for the case of symmetrical single- and double-buffered MINs. It provides accurate results and converges very quickly. The obtained results were validated by extensive simulations and were compared to existing related work in the literature.**

***Index Terms*— multistage interconnection networks, Banyan networks, blocking, performance analysis, switching networks**

I. Introduction

Multistage interconnection networks (MINs) are used as an efficient interconnection medium for multiprocessors, interconnection processors, and memory modules. The behavior of the interconnection networks plays an important role in the performance of multiprocessors. Therefore, to ensure an optimal design, it is necessary to analyze various configurations and constraints of the interconnection networks. A trade-off has to be made between a MIN's complexity and the performance reduction caused by conflicts that might occur when two or more tasks occur simultaneously. However, MINs remain a satisfactory communication medium for parallel systems, in general.

MINs are also a significant component of high speed networks such as Asynchronous Transfer Mode (ATM) networks. In the industry there are several commercial routers which are based on the multistage interconnection networking fabric (e.g. the new CRS-1 Cisco Router [9]).

The performance evaluation of a MIN is of crucial importance. Thus, a lot of research has been devoted to the study of how these networks perform under various conditions through analytical or simulation methods [2, 4–6, 8, 14, 15]. Detailed results can be found for specific cases of MINs which rely mainly on approximation methods for example [2, 8, 24].

The numerical simulation model is based on an analysis of the discrete time behavior of the system. In this case, a formula was derived from analysis of the formula that was extracted by considering the steady state of the MIN. The steady state describes the MIN situation in which the probability of staying in a particular state will not change. In contrast, the classic simulation model determines the system state of each state at each time slot. For instance, it determines how many packets are in a specific queue. If both simulation and mathematical modeling are feasible, then the optimum technique depends on the kind of investigation performed. The mathematical modeling method is a better choice when a lot of tests are required. While a numerical model is time-consuming to create, it can then be used to generate results quickly.

In this paper the focus is on a new analytical method which involves queuing theory, and moreover, is used as a simulator for results validation. So far, several MIN architectures have been proposed in the literature and a lot of work has been devoted to the study and evaluation of the MIN's performance.

The following showcases some of the previous work which has been taken into consideration in the search for a new approach. Most of the MIN analysis focuses on uniform traffic (i.e. packages) coming to a network with an equal probability of reaching output [6, 10, 21]. On the other hand, there are numerous non-uniform traffic patterns in real applications that require special treatment. Such non-uniform approximations can be seen in [3, 22].

Manuscript received July 9, 2009; revised November 24, 2009; accepted February 15, 2010

doi:10.4304/jcm.5.3.247-261

The initial approach in studying MINs mainly considered the case where packets are lost when they try to enter the next stage. Bouras et al. [6] 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 [13]. Garofalakis et al. [5] analyzed banyan networks with finite buffers and came up with the exact solution of the steady-state distribution of the first stage.

They also approximated the solution for the subsequent stages and presented the exact solution for all stages of MINs with single-buffered switches. This proved the well known formula of [5]. Approximations for the performance of packet switched MINs based on uniform traffic can also be found in [4]. Simulations of detailed contention-based network models (used for predicting parallel performance) are still quite challenging, but, relative to one-processor parallel time in [14], decent speedups have been achieved.

Mathematical approaches [5, 8] were also used as a guide in constructing this new analytical model. In Tutsch and Hommel [2, 22], a system of equations was set up for performance estimation. During the set-up, some rules emerged for building such a system. These rules were created for automatic generation of systems of equations in Tutsch [21, 22], which coped with the multicast performance analysis of MINs consisting of switching elements larger than 2×2. Moreover, we have presented a solution for single buffer size MINs in [18], while in this work a general solution for MINs with finite buffers is given.

In the literature there are a lot of publications that are based on different traffic distribution assumptions. For example, Lin and Kleinrock [15] proposed a model for specific hot spot patterns and uniform traffic. Also, Raja et al. [25] used a simulation approach dealing with two types of traffic: traditional Poisson and self-similar traffic. Koppelman et al. [11] conducted an analysis based on offered traffic that follows geometrical distributed message lengths on finite input buffered banyan networks. In 2006, another new solution using simulation was presented by Vasiliadis et al. [17].

Parallel processing is an efficient form of information processing which emphasizes the exploitation of concurrent events in the computing process. To achieve parallel processing, it is necessary to develop more capable and cost-effective systems. Recently, new MIN designs have been introduced; for example an irregular class of fault tolerant MIN named a New Four Tree (NFT) Network was presented in [12]. Also, in [19] a new class of irregular fault tolerant MIN called Improved Four Tree (IFT) was introduced. Besides this, single-chip parallel processing requires high bandwidth between processors and on-chip memory modules. In [24] a hybrid Mess-of-Tree (MoT) buffered network that combines the MoT network with the area efficient butterfly network was introduced. Finally, in [23] the authors proposed a specific multistage architecture that uses PC-based routers as switching elements. This enables them to build a high-speed, large-size, scalable, and reliable software router. All the abovementioned multistage systems require special treatment in calculating performance evaluation issues.

Furthermore, special solutions have been developed for very concrete problems. One such problem exemplified by [1] related to a new method developed for evaluating the residual broadcast reliability of fault-tolerant MINs.

One weakness of existing analytical methods is that they are strictly for very concrete MIN structures and therefore are difficult to adapt to MIN architecture alterations. A new analytical method must be developed to evaluate the performance of similar MIN architecture's. With the above issues in mind, an attempt was made to create an accurate and reliable calculation method. Furthermore, it has to be easily adapted and applied with small changes in some MINs modifying construction schemas.

In this paper, a novel analytical model of a synchronous MIN with finite buffers is presented where this fabric is implemented to work with a backpressure blocking mechanism. An iterative method is proposed for solving the recurrence relationship that defines the equilibrium state probabilities. Various performance measures are derived from the solution and accurate results are presented.

Our research contributes in the following ways:

1. The proposed analytical method provides more accurate results than simulation experiments which require a more time-consuming process [8, 17, 22].
2. In addition, our analytical method converges in a smaller number of iterations than previous ones (e.g. [20]); less than 60 iterations is enough to ensure accurate results.
3. The proposed performance analysis of MINs is robust and flexible. As such, this analysis includes all metrics sufficiently and accurately given various network sizes and buffer length configurations. This has the effect of making their study more detailed and efficient.
4. A 'combined performance factor' for a multi-criteria evaluation of MINs is defined.
5. This methodology is going to act as the basis for the calculation evaluating the performance of MINs in special modern MIN construction alterations. Using this methodology, the more complicated subject of networks as MINs with priorities or MINs which support multicast traffic, or even combinations of them, can be better understood.

The easy adaptation of this analytical approach constitutes its sovereign advantage, particularly compared with Markovian analytical methods that can face more limited breadth in modern complicated MIN performance evaluation issues.

The remainder of this paper is organized as follows: In Section 2, all the required definitions and lemmas of our analysis are given as well as the first level MIN's analytical approximation scheme. In Section 3, the approximate analytical formulae for evaluating the performance of MINs are presented (MINs are exemplified through single- and double-buffered 2 × 2 switching elements). Section 4 provides some of the numerical results generated by our analytical approximation model. These were in turn compared with the results obtained by the simulation experiments. In section 5 the 'combined performance factor' is defined and in section 6 the methodology's expandability is discussed. Finally, in Section 7, our conclusions and anticipated future work are presented.

II. PROPOSED APPROXIMATE ANALYTICAL MODEL

A. MIN analysis

In general, an $N \times N$ MIN is constructed from $L = \log_k N$ stages of $k \times k$ Switching Elements (SEs), where k is the degree of the SEs. Let (i) depict an arbitrary number of stages, where (i) can be escalated from 1 to L. Generally, each SE consists of k -input and k -output ports. In the fabric, there are exactly (N/k) SEs at each stage, so the total number of SEs of a MIN is $((N/k)\cdot \log_k N)$ (Fig. 1). There are $(N \cdot \log_k N)$ interconnections among all stages, unlike the crossbar network, which requires $O(N^2)$ SEs and links. There is a unique path from each processor (source node) to each memory module (sink node), and therefore the studied MIN belongs to the class of Banyan Networks (BNs). 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 (Fig. 1). In each output port there is a buffer. We assume that the buffers may be of finite or zero length (single- or double-buffered switches). Such a network can be modeled as a labeled 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 labeled digraph each edge represents one or more lines going from a node to its successor.

The whole network operates 'synchronously', which means that the time cycles refer to global clock ticks. The network clock cycle consists of two phases. In the first phase, flow control information passes through the network from the last stage to the first stage. In the second phase, packets flow from one stage to the next in accordance with the flow control information.

The routing algorithm applied here, assumes that there is a fixed path which has to be followed by a packet throughout the network. The path can be encoded as a sequence of labels of the successive switch outputs of the path (path descriptor). More concretely, the SEs in multistage networks are digit-controlled crossbars. This is done by including a control sequence in the packet, named a packet control sequence. The control sequence is a series of digits allocated for each stage of the network. The digit indicates which output of the SE is to be connected to the input. Therefore, the control sequence represents the path to be taken by the message through the MIN. Packets are generated at each processor by independent, identically distributed random processes. In this analysis it is assumed 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 (First Input First Output) policy for their servers (outputs). Conflicts between packets simultaneously routed to the same output port are resolved by queuing the packet.

Our analysis assumes that packets moving from stage i to stage $(i+1)$ and finding the output buffer of stage $(i+1)$ full will block the server (output) of their origin's output (stage i). The above, of course, does not apply to the processor's feeding stage $i = 1$ (i.e. (p) remains the same in every cycle) or to the buffers of the last stage $i = L$, which are not supposed to become blocked under any condition. Blocking of an output is interpreted as stopping its operation, that is, it cannot accept any packets for service (it cannot forward packets to the next stage).

The service time of the output queues of each switch is assumed to be 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, each with a deterministic server (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.

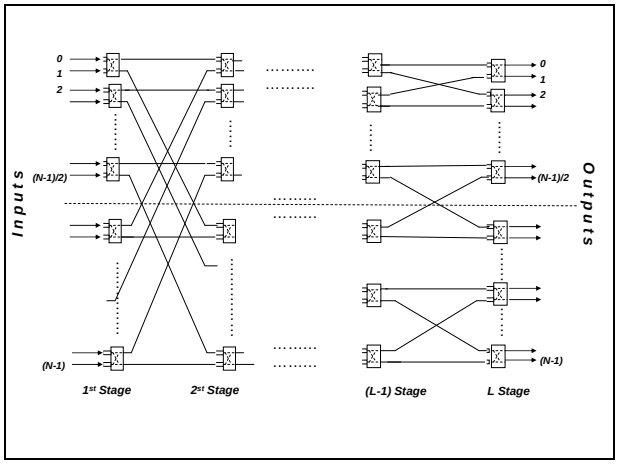

Figure1An $N \times N$ single buffered MIN (Delta type) with L stages constructed by SEs with k=2

In our analysis we assume that the buffer length (b) does not include the server (output). So, a single-buffered switch is assumed 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 there are any). The routing logic at each switch is assumed to be fair, that is, conflicts are randomly resolved. In addition, it is worthy of reminder that our analysis is based on homogeneity, and thus all switches in a stage and, likewise, all outputs within a switch are statistically identical.

B. Basic definitions

Let us consider stages i and $i+1$ (for $i=1$ to $i=L-1$). A particular output queue $OQ^{(i)}$ of stage i will finally be able to send a packet (when it has one) only when it selects a queue of stage $i+1$ which is not full.

Definition 1 Let $u^{(i)}$ be the steady-state probability that a particular output server of stage i of the $k \times k$ switch network is *busy*. An output server is busy either because it is serving a packet or because it is blocked. This is the *utilization* in steady state of an output buffer of stage i of the $k \times k$ switch network. An arbitrary queue of a MIN with buffer size (*b*) has a number (*b*) of possible utilized states. The probability of those distinct queue states is expressed as: $u_j^{(i)}$. The $u_j^{(i)}$ express the queue *utilization* by (*j*) packet population (*j* = 1, 2, ..., *b*). Also, the queue *utilization* is given: $u^{(i)} = \sum_{j=1}^{b} u_j^{(i)}$.

Definition 2 Let $p_b^{(i)}$ be the steady-state probability that a particular output server of stage i of the $k \times k$ switch network is *blocked*. Obviously, $p_b^{(L)} = 0$.

Definition 3 Let $p_{serv}^{(i)}$ be the steady-state probability that a particular output server of stage i of the $k \times k$ switch network is *serving* a packet.

Definition 4 Let $p_0^{(i)}$ be the steady-state probability that a particular output buffer of stage i of the $k \times k$ switch network is empty. Obviously, $u^{(i)} = 1 - p_0^{(i)}$.

Definition 5 Let $C_k^{(i)}$ be the random variable denoting the number of packets *arriving* at an output buffer of a $k \times k$ switch of stage *i* ($i = 1..L$), of the network at the end of a cycle and $x_{k,c}^{(i)} = \Pr(C_k^{(i)} = c)$.

Any queue in the system can be utilized by '*normal*' or *blocked* packets. '*Normal*' packets are the packets that have just arrived in the queue and are ready for service the next time, whereas *blocked* packets are the packets that have already tried to be serviced but have been blocked for any reason and therefore remain in the queue. Thus, the utilization in a queue can be expressed as:

$$u^{(i)} = p_{serv}^{(i)} + p_b^{(i)} \quad (1)$$

C.. Performance metrics

- *Throughput* of a MIN, T_h, is defined as the number of packets delivered to their destinations per unit of time. Nevertheless, because the queues of the last stage are never blocked, the *utilization* of the last stage queues is equal to the MIN's *throughput*. So, $u^{(L)} = T_h$.
- *Normalized throughput* of a MIN, T_{hN}, is defined as a ratio of the *average throughput* T_h to the network size N. Formally, T_{hN} is expressed by $T_{hN} = \frac{T_h}{N}$.
- *Average latency* $\overline{D}$ is the average time a packet spends passing through the MIN. Formally, $\overline{D}$ is expressed by

$$\overline{D} = \lim_{t \to \infty} \frac{\sum_{i=1}^{n(t)} d(i)}{n(t)} \quad (2)$$

where *n(t)* denotes the total number of packets accepted within *t* time slots and *d(i)* represents the total delay for the i^{th} packet. Recall from Section 2 that the packets are routed by store and forward routing from one stage to the next by the backpressure mechanism. The *d(i)* is considered to be the sum of q_i and t_i, where q_i denotes the total queuing delay for the i^{th} packet waiting at each stage and t_i denotes the total transmission delay of the i^{th} packet at each stage of the MIN.

Consequently, taking into account that all queues are single-buffered, the average delay of a packet traversing the network can be calculated by $\sum_{i=1}^{L} u^{(i)}$, whereas the average probability of a packet being accepted n a queue of the first stage is equal to the *utilization* $u^{(L)}$ of the last stage L, because no packets are lost within the network and all packets are removed from their destinations immediately after arrival. Thus Equation (2) can be simplified to $\overline{D} = \frac{\sum_{i=1}^{L} u^{(i)}}{u^{(L)}}$.

- *Normalized latency* $\overline{D}_N$ of packets traversing a MIN with *L* stages can be defined as the ratio of the *average latency* $\overline{D}$ of packets to the *minimum delay* required by a packet to traverse the MIN without any blocking. This *minimum delay* depends on the number of stages that have a MIN. So, the *normalized latency* $\overline{D}_N$, can be expressed by the formula

$$\overline{D}_N = \frac{\overline{D}}{L} \quad (3)$$

where $\overline{D}$ is the *average latency* of packets traversing the MIN.

D. First level approximation scheme

The number of cycles needed for the output queue $OQ^{(i)}$ to successfully send a packet after *j* trials can be

approximated by $\Pr(OQ^{(i)} = \left(u^{(i+1)}\right)^{j-1}\left(1-u^{(i+1)}\right)$; that is, $OQ^{(i)}$ operates with a geometric service time process of success probability $1-u^{(i+1)}$.

In the spirit of the approximation for solving the stages beyond the first in Bouras et al. in [5], let us assume that stage $i-1$ behaves as processors with packet generation probability $u^{(i-1)} = 1-p_0^{(i-1)}$. The boundary conditions are $u^{(0)} = p$ for the processors (stage 0) feeding the first stage of the MIN, and $u^{(L+1)} = u^{(L)}$ for the destination of the packets beyond the MIN (i.e. the last stage is never blocked).

Because of the above assumptions, we now have:

Remark A. A particular output queue of stage i (for $i=1$ to $i=L-1$) can be approximated by a discrete queue of size *b*, of geometric service time, with exit probability $(1-u^{(i+1)})$, and of bulk arrivals, where the number of arrivals in any cycle is a Bernoulli of *k* trials and success probability $(1-p_0^{(i-1)})/k$. Let us call such a queue a *Be/G/1/b*. For the last stage *L*, the queues are *Be/D/1/b*, that is, the service time is assumed to be constant and equal to the network cycle time (with value 1), since the last stage is never blocked.

Notice, however, that in the general case, in order to get the parameters of the arrival process and service time of a queue at stage i, one has to know the solutions of stages $(i+1)$ and $(i-1)$.

Thus, our approximation scheme is now a convergence procedure where the following two phases are repeated until the queue utilizations do not change any more. The scheme is initialized by letting all queue utilizations be equal to 0.

Iterative algorithm

PHASE A (backward solution of the MIN). Starting from the last stage *L*, solve for $u^{(L)}$ to get the parameter of the geometric service process of stage $(L-1)$. Repeat until stage 1 is reached.

PHASE B (forward solution of the MIN). Starting from the first stage, with input parameter *p* and geometric service as found from phase A, find its utilization. Use this as input parameter for stage 2, and so on, until the last stage is reached.

In Sections 4, 5 and in Appendix A, we present this mathematical convergence method for the single- and double-buffered MINs with variable network sizes.

E. Additional Definitions

For the general case of an L-stage MIN, consisting of $k \times k$ switches, with output buffers of length b ($b < \infty$) in all stages, we have:

Definition 6 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 being generated by a processor in each cycle. Therefore:

$$x_{k,c}^{(1)} = \begin{cases} \binom{k}{c}\left(\frac{p}{k}\right)^c\left(1-\frac{p}{k}\right)^{k-c}, & for\ 0 \le c \le k \\ 0, & otherwise \end{cases} \quad (4)$$

Definition 7 The arrival process of packets at the output queues of stage i (for $i=2$ to $i=L$) of the network, is approximated by a binomial distribution *bin(*k, $u^{(i-1)}/k$*)*, where $u^{(i-1)}$ is the utilization of an arbitrary queue of stage $i-1$, which we assume to play the role of the fixed probability of packets which are generated by processors at each cycle, feeding stage *i*. Therefore:

$$x_{k,c}^{(i)} \approx \begin{cases} \binom{k}{c}\left(\frac{u^{(i-1)}}{k}\right)^c\left(1-\frac{u^{(i-1)}}{k}\right)^{k-c}, & for 0 \le c \le k \ and\ 2 \le i \le L \\ 0, & for all other values\ of\ c \end{cases} \quad (5)$$

Definition 8 The *state* of an arbitrary output queue of stage *i* at the end of cycle *n* is a two-dimensional variable, with $2b+1$ possible values: *{(0,0), (1,0), (2,0) ..., (b,0), (1,1), (2,1), ..., (b,1)}*, where in *(x,y) x* is the number of packets in the output buffer, and *y* can take two values: 0 when the output queue is not blocked, or 1 when it is.

Definition 9 Let $(q,s)_k^{(i)(n)}$ be the random variable denoting the state of an arbitrary output queue of stage *i* at the end of cycle *n*, where *q* is the number of packets in the output buffer and *s* is 0 if the output queue is not blocked, or 1 when it is. Let $(q,s)_k^{(i)}$ be the steady-state limit of $(q,s)_k^{(i)(n)}$.

Definition 10 Let $v_k^{(i)(n)}$ be the number of packets that *are entering* an arbitrary output queue of stage *i* at the end of cycle *n*, and let $v_k^{(i)}$ be the steady-state limit of $v_k^{(i)(n)}$. It holds that $v_k^{(i)(n)} \le C_k^{(i)}$ at each cycle *n*.

Definition 1 Let $p_{k,q,s}^{(i)} = \Pr[(q,s)_k^{(i)} = (q,s)]$, $0 \le q \le b$, $s=0$ when the queue is not blocked and 1 when it is blocked be the distribution of $(q,s)_k^{(i)(n)}$ in the steady state. So, $u^{(i)} = 1-p_{k,0,0}^{(i)} = 1-p_0^{(i)}$ is the utilization of an arbitrary queue of stage *i*.

H. Lemmas

Lemma 1 For $0 < m \le \min(b,k)$ and for all stages *i* of the network: For *q*, *s* from $(q,s)_k^{(i)(n-1)}$ it holds that:

$$\Pr(v_k^{(i)(n)} = m) = \begin{cases} x_{k,m}^{(i)} & if\ q-\Delta(q)+\Delta(s)+m < b \\ x_{k,m}^{(i)} + x_{k,m+1}^{(i)} + \cdots + x_{k,k}^{(i)} & if\ q-\Delta(q)+\Delta(s)+m = b \\ 0 & otherwise \end{cases} \quad (6)$$

where: $\Delta(q) - \Delta(s)$ is the departure of a packet from an arbitrary output queue of stage *i* at the end of cycle *n* if there is a packet and if the output server is not blocked.

For $m = 0,\ \Pr(v_k^{(i)(n)} = 0) = x_{k,0}^{(i)}$, for any q.

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

The proof of this lemma is similar to the proof of the related lemma in Bouras et al. [5]. In summary, it states that an output buffer of stage (i) will accept as many packets as there are vacancies in the buffer. The remaining packets will block their origin's output queues.

Lemma 2 *Relating blocking probabilities with utilization***.** In a MIN with blocking, for all stages except the last one, the *probability of blocking* in stage *i* (where *i* = 1 … (*L* – *1*)) is equal to the difference in the probabilities of *utilization* in stage *i* and the *utilization* in the last stage (*L*).

$$p_b^{(i)} = u^{(i)} - u^{(L)} \tag{7}$$

Proof In every queue except the queues of the last stage we have from Equation (1):

$$u^{(i)} = p_{serv}^{(i)} + p_b^{(i)} \tag{8}$$

We use the following operational argument:

Let $S^{(i)}$ be the total service time spent in stage (i) by all packets traversed through the MIN; i = 1, ..., L; that is $S^{(i)} = N_T$, where N_T is the total number of packets generated during *T* which were not lost on entering stage 1, since service time = 1. Due to homogeneity, for a queue of stage (i), the total service time is $x^{(i)} = \frac{N_T}{M}$, where M is the number of input ports to the MIN.

Thus, $p_{serv}^{(i)} = \frac{x^{(i)}}{T} = \frac{N_T}{M \cdot T}$ for *i* = 1, …, *L – 1*.

So, $u^{(i)} = p_{serv}^{(i)} + p_b^{(i)} = \frac{x^{(i)} + b^{(i)}}{T}$, where $b^{(i)}$ is the time during which the queue is blocked. For the last stage, $u^{(L)} = \frac{x^{(L)}}{T} = \frac{x^{(i)}}{T} = p_{serv}^{(i)}$, where *i* = 1, 2, ..., *L*.

So, because all the entering packets in the MIN are not lost:

$$p_{serv}^{(1)} = p_{serv}^{(2)} = ... = p_{serv}^{(L)} \tag{9}$$

But because in the last stage we do not have blocking, $p_{serv}^{(L)} = u^{(L)}$, and

$$u^{(i)} = u^{(L)} + p_b^{(i)};\ \text{thus}\ p_b^{(i)} = u^{(i)} - u^{(L)} \tag{10}$$

III. THE APPROXIMATE SOLUTION FOR MIN WITH 2×2 SES

A. The general approximate solution for b-buffered MIN with 2x2 SEs

In order to demonstrate the approximation scheme and the underlying analytical assumptions and techniques, we start by applying the scheme to the general case, with MINs consisting of 2×2 switches, with finite buffer size (b). Let state (*s)* represents the state of an arbitrary queue of MIN when its packet population is equal to *s* (where *s*=0...*b*). The total number of possible states for each queue is (*b*+1). These distinct states of an arbitrary *i*-stage are denoted by the probabilities: $p_0^{(i)}$ and $u_s^{(i)}$.

- $p_0^{(i)}$ is the queue probability of being empty, and
- $u_s^{(i)}$ is the *utilization* of queue when it holds (s) number of packets (where s =1..*b)*.

Consequently, the aggregate *utilization* of an *i*-stage queue is given by: $u^{(i)} = \sum_{s=1}^{b} u_s^{(i)}$. The aggregate probability of all states is: $p_0^{(i)} + \sum_{s=1}^{b} u_s^{(i)} = 1$ (11)

According to lemma 1, (*b*+1) equations can be applied; one for each distinct state, providing the following system of equations:

$$\begin{aligned}
p_0^{(i)} &\approx p_0^{(i)} \cdot x_{20}^{(i)} + u_1^{(i)} \cdot (1 - p_b^{(i+1)}) \cdot x_{20}^{(i)} \\
u_1^{(i)} &\approx p_0^{(i)} \cdot x_{21}^{(i)} + u_1^{(i)} \cdot [p_b^{(i+1)} \cdot x_{20}^{(i)} + (1 - p_b^{(i-1)}) \cdot x_{21}^{(i)}] + u_2^{(i)} \cdot (1 - p_b^{(i-1)}) \cdot x_{20}^{(i)} \\
u_2^{(i)} &\approx p_0^{(i)} \cdot x_{22}^{(i)} + u_1^{(i)} \cdot [p_b^{(i)} \cdot x_{21}^{(i)} + (1 - p_b^{(i+1)}) \cdot x_{22}^{(i)}] + u_2^{(i)} \cdot [p_b^{(i+1)} \cdot x_{20}^{(i)} + (1 - p_b^{(i+1)}) \cdot x_{21}^{(i)}] + u_{b3}^{(i)} \cdot (1 - p_b^{(i+1)}) \cdot x_{20}^{(i)} \\
u_3^{(i)} &\approx u_1^{(i)} \cdot [p_b^{(i+1)} \cdot x_{22}^{(i)}] + u_2^{(i)} \cdot [p_b^{(i+1)} \cdot x_{21}^{(i)} + (1 - p_b^{(i+1)}) \cdot x_{22}^{(i)}] + u_3^{(i)} \cdot [p_b^{(i+1)} \cdot x_{20}^{(i)} + (1 - p_b^{(i+1)}) \cdot x_{21}^{(i)}] + u_4^{(i)} \cdot (1 - p_b^{(i+1)} \\
&\ldots\ldots\ldots\ldots \\
u_{(b-1)}^{(i)} &\approx u_{(b-3)}^{(i)} \cdot [p_b^{(i+1)} \cdot x_{22}^{(i)}] + u_{(b-2)}^{(i)} \cdot [p_b^{(i+1)} \cdot x_{21}^{(i)} + (1 - p_b^{(i+1)}) \cdot x_{22}^{(i)}] + u_{(b-1)}^{(i)} \cdot [p_b^{(i+1)} \cdot x_{20}^{(i)} + (1 - p_b^{(i+1)}) \cdot x_{21}^{(i)}] + u_b^{(i)} \cdot \\
u_{(b)}^{(i)} &\approx u_{(b-2)}^{(i)} \cdot [p_b^{(i+1)} \cdot x_{22}^{(i)}] + u_{(b-1)}^{(i)} \cdot [p_b^{(i+1)} \cdot (x_{21}^{(i)} + x_{22}^{(i)}) + (1 - p_b^{(i+1)}) \cdot x_{22}^{(i)}] + u_{(b)}^{(i)} \cdot [p_b^{(i+1)} \cdot x_{20}^{(i)} + (1 - p_b^{(i+1)}) \cdot (x_{21}^{(i)} +
\end{aligned} \tag{12}$$

In fact, the above system (12) of $(b+1)$ equations is a linear and homogenous system. Combining the first (*b*) equations of the system (12) with the equation (11) forms a new linear system (but not homogenous) of (*b+1*) equations with (*b+1*) unknowns. This general system of $(b+1)$ equations (for buffer size=*b*) has the following linear structure:

$$\begin{array}{lllllll}
p_0^{(i)} \cdot a_{11} + & u_1^{(i)} \cdot a_{12} + & u_2^{(i)} \cdot 0 + & u_3^{(i)} \cdot 0 + & u_4^{(i)} \cdot 0 + & \ldots + u_b^{(i)} \cdot 0 & \approx 0 \\
p_0^{(i)} \cdot a_{21} + & u_1^{(i)} \cdot a_{22} + & u_2^{(i)} \cdot a_{23} + & u_3^{(i)} \cdot 0 + & u_4^{(i)} \cdot 0 + & \ldots + u_b^{(i)} \cdot 0 & \approx 0 \\
p_0^{(i)} \cdot a_{31} + & u_1^{(i)} \cdot a_{32} + & u_2^{(i)} \cdot a_{33} + & u_3^{(i)} \cdot a_{34} + & u_4^{(i)} \cdot 0 + & \ldots + u_b^{(i)} \cdot 0 & \approx 0 \\
p_0^{(i)} \cdot 0 + & u_1^{(i)} \cdot a_{42} + & u_2^{(i)} \cdot a_{43} + & u_3^{(i)} \cdot a_{44} + & u_4^{(i)} \cdot a_{45} + & \ldots + u_b^{(i)} \cdot 0 & \approx 0 \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots\ \ldots & \ldots\ldots \\
p_0^{(i)} \cdot 0 + & u_1^{(i)} \cdot 0 + & u_2^{(i)} \cdot 0 + & u_3^{(i)} \cdot 0 + & u_4^{(i)} \cdot 0 + & \ldots + u_b^{(i)} \cdot a_{(b-1)b} & \approx 0 \\
p_0^{(i)} \cdot 1 + & u_1^{(i)} \cdot 1 + & u_2^{(i)} \cdot 1 + & u_3^{(i)} \cdot 1 + & u_4^{(i)} \cdot 1 & \ldots + u_b^{(i)} \cdot 1 & \approx 1
\end{array} \tag{13}$$

All coefficients a_{ij} are expressions of $p_b^{(i+1)}$ and $x_{2,j}^{(i)}$ where:

- $p_b^{(i+1)}$ is the *blocking probability* of the successive stage and
- $x_{2,j}^{(i)}$ is the probability of *packet arrivals* at the current stage, where *j*={0,1,2} denotes the *probability of that j* packets arrive from the previous stage.

Thus, all the coefficients a_{ij} are expressions, i.e. $a_{11} = x_{2,0}^{(i)} - 1$, $a_{12} = (1 - p_b^{(i+1)}) \cdot x_{2,0}^{(i)}$, $a_{21} = x_{2,1}^{(i)}$, etc. Furthermore:

- According to definition 8, the *packet arrivals* $x_{2,j}^{(i)}$ can be expressed as a function of the *utilization* of the precedent stage $x_{2,j}^{(i)} = f_j(u^{(i-1)})$

- According to lemma 2, the *blocking probability* $p_b^{(i+1)}$ of the successive stage (i+1) is also a function of the *utilization* according to the lemma 2.

Thus, all the coefficients a_{ij} are a *utilization* function: $a_{ij} = f_u(u^{(i-1)}, u^{(i+1)}, u^{(L)})$ That means all factors include exclusive *utilization* metrics.

The above system of equations (13) can be solved by applying Cramer's theorem, as follows:

$p_0^{(i)} = \frac{D_0^{(i)}}{D^{(i)}}$, and similarly, all other state probabilities can be estimated by:

$$u_s^{(i)} = \frac{D_s^{(i)}}{D^{(i)}} \text{ , for } s=1...b \qquad (14)$$

Where: $D^{(i)}$, $D_0^{(i)}$ and $D_s^{(i)}$ are Cramer's matrices.

The aggregate queue *utilization* can be calculated by:

$$u^{(i)} = 1 - p_0^{(i)} = 1 - \frac{D_0^{(i)}}{D^{(i)}} \text{ , } \qquad (15)$$

The formula (15) is in fact a recursive formula because both matrices, $D^{(i)}$, and $D_0^{(i)}$, include only *utilization* metrics. In particular, the *utilization* of the previous, current, successive and last stage queues are included. Thus, $D_0^{(i)} = f(u^{(i-1)}, u^{(i)}, u^{(i+1)}, u^{(L)})$, and $D^{(i)} = f(u^{(i-1)}, u^{(i)}, u^{(i+1)}, u^{(L)})$, and then the aggregate utilization of an i-stage queue is: $u^{(i)} = f(u^{(i-1)}, u^{(i)}, u^{(i+1)}, u^{(L)})$.

This is the reason why an iterative algorithm is used for approaching the solution of the general recurrence relationship (15). The convergence of this recursive algorithm will define the equilibrium state *utilization's* probabilities. Thus, applying this convergent algorithm (which is demonstrated in the following section, 4.2), a convergence at a fixed point is required.

In order to evaluate the probabilities above, we make the assumption of *approximate interstage independence* (which seems to be more accurate, as b is getting smaller). Actually, Kruskal and Snir in [13] derive the same equation for $u^{(i)}$, i = 1, 2,.., L, as we do, for the single buffered MIN without blocking (a case clearly with interstage independence), giving evidence that our assumption is approximately true for small b, when packets are lost. In our case of blocking, there is of course a stronger dependence among stages, which is taken into account in some extent by adopting Remark A. Comparison to simulation results later here, show that this assumption is a reasonable one.

Boundary conditions: The requirements for the first and last stage are as follows:

- For the first stage, $i = 0$:

Since there is no preceding stage, the *probability of packet arrivals to the inputs* (p), is the offered load to the network inputs. So, $u^{(0)} = p$

- For the last stage, $i = L$:

A packet at an output port of the last stage can always proceed. However, buffers in the SEs of the last stage can not proceed in the blocked state. Thus: $p_b^{(L)} = 0 \Leftrightarrow u^{(L+1)} = u^{(L)} = p_{serv}^{(L)}$

The convergence algorithm: Using a fixed-point iteration ($\varepsilon < 10^{-4}$) over the state *utilization*, a steady state is reached from which performance metrics of interest are determined. The convergent algorithm is presented in Appendix A. The illustrated algorithm in Appendix A includes the formulas for the *utilization*. Evaluation of the *blocking probabilities* can be calculated similarly.

B. Case Studies: Single and double buffered MINs

B.1 Approximate solution for single buffered MIN with 2x2 SEs

The above demonstrated approximate convergent method is exemplified here for MINs, consisting of 2×2 single buffered ($b = 1$) SEs. The steady-state distribution in this case consists of two distinct states: $p_0^{(i)}$, the probability of the queue being empty and $u^{(i)} = u_1^{(i)}$, is the probability that the output queue utilizes a packet.

When we have a small buffer size, the above general equation (15) has a solution which is expressed by a closed formula, as shown here.

Using the analysis derived from the aforementioned sub-section 4.1 (which is based on lemma 1), it can be approximated by the following:

$$\left.\begin{aligned} p_0^{(i)} &\approx p_0^{(i)} \cdot x_{2,0}^{(i)} + u^{(i)} \cdot (1 - p_b^{(i+1)}) \cdot x_{2,0}^{(i)} \\ 1 &\approx p_0^{(i)} + u^{(i)} \end{aligned}\right\} \qquad (16)$$

By solving the above set of equations (16), we get:

$$u^{(i)} = 1 - p_0^{(i)} \approx \frac{1 - x_{2,0}^{(i)}}{1 - x_{2,0}^{(i)} \cdot p_b^{(i+1)}} \qquad (17)$$

Using equation (3) of definition (8), we arrive at: $x_{2,0}^{(i)} \approx (1 - \frac{u^{(i-1)}}{2})^2$ (utilization's expression) and applying lemma 2 for $(i+1)$ stage, we arrive at: $p_b^{(i+1)} = u^{(i+1)} - u^{(L)}$ (utilization's expression)

For stage (i), the above equations (17), can be replaced by:

$$u^{(i)} \approx \frac{1 - (1 - \frac{u^{(i-1)}}{2})^2}{1 - (1 - \frac{u^{(i-1)}}{2})^2 \cdot (u^{(i+1)} - u^{(L)})} , \qquad (18)$$

The formula (18) (also known as the utilization's expression) is the recursive formula for an single buffered case of MIN. For stages $i=1$ and $i=L$, the same formula (18), using the relevant boundary conditions, is used. Boundary Conditions: remains the same as is suggested in sub-section 4.1, above.
The convergence algorithm: is the same algorithm as is demonstrated above in sub-section 4.2. However, the current formula (18) is used to demonstrate this special case study, instead of the general formula (15).

B.2 Approximate solution for double buffered MIN with 2x2 SEs

Also the above demonstrated approximate convergent method is exemplified here for MINs, consisting of 2×2 double buffered ($b=2$) SEs.

The steady-state distribution in this case consists of two distinct states:

$p_0^{(i)}$, the probability of the queue being empty,

$u_1^{(i)}$, is the probability that the output queue utilizes a packet and

$u_2^{(i)}$,is the probability that the output queue utilizes two packets. Thus, $u^{(i)}=u_1^{(i)}+u_2^{(i)}$

Using the analysis derived from the aforementioned sub-section 4.1 (which is based on lemma 1), it can be approximated by the following:

$$\left.\begin{array}{llll} p_0^{(i)}\cdot(1-x_{2,0}^{(i)})-u_1^{(i)}\cdot(1-p_b^{(i+1)})\cdot x_{2,0}^{(i)} & +u_2^{(i)}\cdot 0 & =0 \\ p_0^{(i)}\cdot x_{2,1}^{(i)} \quad +u_1^{(0)}\cdot[p_b^{(i+1)}\cdot x_{2,0}^{(i)}+(1-p_b^{(i+1)})\cdot x_{2,1}^{(i)}-1]+u_2^{(i)}\cdot(1-p_b^{(i+1)})\cdot x_{2,0}^{(i)} & & =0 \\ p_0^{(i)} \quad +u_1^{(i)} & +u_2^{(i)} & =1 \end{array}\right\} \quad (19)$$

Setting on the system:

$A=p_b^{(i+1)}\cdot x_{2,0}^{(i)}+(1-p_b^{(i+1)})\cdot x_{2,1}^{(i)}-1$,

$B=(1-p_b^{(i+1)})\cdot x_{2,0}^{(i)}$

By solving the above set of equations (19), we get:

$$D_{p_0^{(i)}}=\begin{vmatrix} 0 & -B & 0 \\ 0 & A & B \\ 1 & 1 & 1 \end{vmatrix}=-B^2 \text{ and } D=\begin{vmatrix} (1-x_{2,0}^{(i)}) & -B & 0 \\ x_{2,1}^{(i)} & A & B \\ 1 & 1 & 1 \end{vmatrix}$$

$$=(1-x_{2,0}^{(i)})\cdot(A-B)+x_{2,1}^{(i)}\cdot B-B^2$$

$$p_0^{(i)}=\frac{D_{p_0^{(i)}}}{D}=\frac{-B^2}{(1-x_{2,0}^{(i)})(A-B)+x_{2,1}^{(i)}\cdot B-B^2}$$

Where: $x_{2,0}^{(i)}$, $x_{2,1}^{(i)}$, A, B and $p_b^{(i+1)}$ are *utilization's* expressions. Thus,

$$p_0^{(i)}=\frac{-(1-p_b^{(i+1)})^2\cdot(x_{2,0}^{(i)})^2}{(1-x_{2,0}^{(i)})\cdot[(p_b^{(i+1)}x_{2,0}^{(i)}+(1-p_b^{(i+1)})\cdot x_{2,1}^{(i)}-1)-(1-p_b^{(i+1)})x_{2,0}^{(i)}]+x_{2,1}^{(i)}x_{2,0}^{(i)}(1-p_b^{(i+1)})-(x_{2,0}^{(i)})^2(1-p_b^{(i+1)})^2}$$

And finally, $u^{(i)}=1-p_0^{(i)}$ (20)

The formula (20) (also a utilization's expression) is the recursive formula for the double buffered case of MIN.

Boundary conditions: remains the same as is suggested in sub-section 4.1, above.

The convergence algorithm: is the same algorithm as is demonstrated in Appendix A. However, the current formula (20) is used to demonstrate this special case study, instead of the general formula (15).

IV. Applying the Arithmetic Convergent Method and Simulation

This arithmetic convergent method is presented and exemplified through its application on a MIN consisting of 2 × 2 single- or double-buffered (b = 1 or 2) SEs under various network sizes. A number of different experiments were performed. In each experiment, less than 60 iterations were used to achieve a convergence ($\varepsilon<10^{-4}$). The probability (p) of packets arriving at the inputs of the MIN ranked from 0.2 to 1.

A simulator was also constructed and applied under the same MIN conditions in order to validate the results given by the analytical method. All performance metrics obtained from the simulation ran for 10^5 clock cycles. The number of simulation runs was adjusted to ensure a steady-state operation condition for the MIN.

All performance metrics such as utilization, latency, service, and blocking probabilities were recorded for each queue in all stages. It is clear that all statistics obtained by simulation experiments verified the numerical results of our novel approximate mathematical solution. Also, all the data that are presented in sub-sections 5.2 and 5.3 are evaluated using the Relative Statistical Error (RSE) indicator.

The definition of RSE is: $RSE=\frac{\Delta y}{y}\cdot 100\%$.

The difference between the measured or inferred value of a performance quantity y_0 and its actual value y is given by $\Delta y=y_0-y=t_{n-1,1-\varepsilon/2}\cdot\sqrt{s^2/n}$ where

- y is the performance metric under study (e.g. normalized throughput, average packet latency),
- ε gives the expected error (here, $\varepsilon=10^{-3}$),
- s^2 expresses the variance of a finite number of values, and
- $t_{n-1,1-\varepsilon/2}$ gives the quantile of the t-distribution with $n-1$ degrees of freedom.

The RSE gives the absolute statistical error. The estimated RSE is closely related to the confidence level. Our simulations were performed using an accuracy of 10^{-3} and RSEs in all cases of our experiments were less than 4% ($RSE<4\%$).

A. Model verification

1) Compare normalized throughput

The proposed novel analytical model was also validated by four older classic models: Jenq's model [7],

Mun's model [16], Theimer's model [20], and Yoon's model.

i) Compare the results of a single-buffered MIN.

The proposed novel analytical model was also validated by three older classic models: Jenq's model [7], Mun's model [16], and Theimer's model [20].

Figure 2 depicts the normalized throughput of a six-stage single-buffered MIN (64 × 64) versus the offered load. It is worth noting that all models are accurate at low loads, but their accuracy decreases as the packet arrivals at inputs increase. According to Figure 2, the accuracy of Jenq's model is less sufficient under moderate and high traffic conditions (p > 0.4) because many packets are blocked, mainly at the first stages of the MIN, especially at high traffic rates.

Mun's model improves the accuracy by introducing a 'blocked' state. Moreover, Theimer et al. introduce the dependencies between the two buffers for each switching element, improving their model and approaching the simulation experiments better than Mun's model.

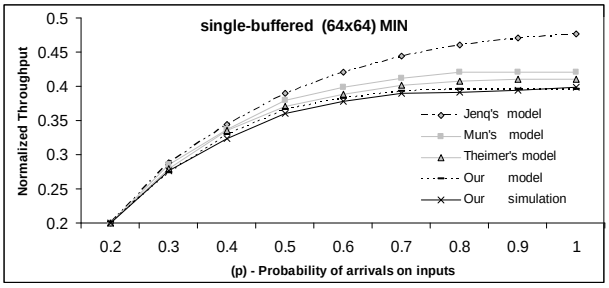

Figure 2. Normalized throughputs versus probability of packet arrivals for a six-stage MIN

Finally, our novel analytical method achieves better approaches than all previous models (Fig. 2) using a very fast convergence (less than 60 iterations).

ii) Compare the results of a double-buffered MIN.

The proposed analytical model was also validated by two older classic models: Mun's model [16] and Yoon and colleagues' model [25]. Figure 3 depicts the normalized throughput of a six-stage double-buffered MIN (64 × 64) versus the offered load. It is worth noting that all models are accurate at low loads, but their accuracy decreases as the packet arrivals at inputs increase.

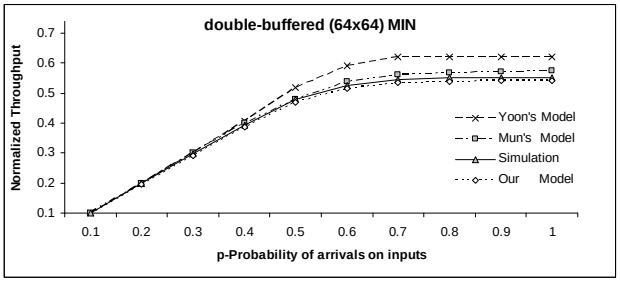

Figure 3. Normalized throughputs versus probability of arrivals at inputs in a six-stage double-buffered MIN

The plots clearly verify that our model is more accurate than the other two models. The Yoon model is the worst case since it does not consider the blocked state and the rest of its assumptions are simple. Also, Mun's model gives less accurate results owing to the probabilistic complexity of the model. Both models give a *throughput* overestimation in final stages. That overestimation happens because in the later stages both models' calculated values of blocking probabilities underestimate their real values. Actually, with high traffic, many packets can be blocked even from the first stage.

In conclusion, comparisons with other existing models revealed that the proposed model is considerably more accurate, irrespective of the network size, buffer size, or offered load.

Finally, according to Figures 2 and 3 the *normalized throughput* of a six-stage MIN is close to 40% for single- and 56% for double-buffered MIN configurations respectively, under full offered load conditions. Consequently, the extra buffer availability leads in turn to far fewer blockings, and thus the throughput gain was found to be very significant (40%).

2) Compare average packet latency of single- and double-buffered MIN.

Figure 4 depicts the *average packet latency* of a six-stage MIN (64 × 64) versus the *offered load*. The solid curves illustrate results for the single-buffered case while the dotted curves depict results for the corresponding case of a double-buffered MIN. It is worth noting that all models are accurate at low loads, but their accuracy decreases as the packet arrivals at inputs increase.

According to this figure, the results obtained by our analytical model were in close agreement with those of our corresponding simulation experiments for both configuration set-ups (b = 1, 2), again demonstrating the accuracy of our proposed analytical method.

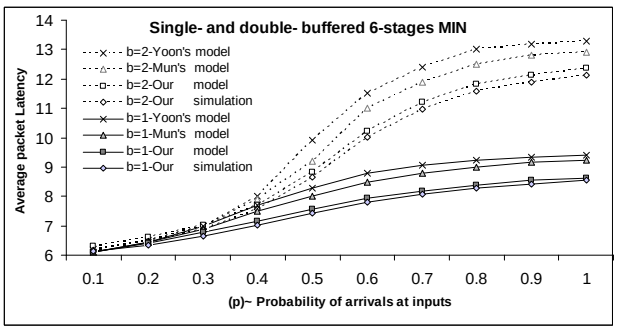

Figure 4. Average packet latency versus probability of packet arrivals for single- and double-buffered six-stage MINs

It is also noticed that using double-buffered queues leads to more delays'. This behavior becomes perceptible even at low loads (p = 0.4), while the delay increment becomes apparent at medium and high loads ($p \geq 0.6$).

B. Performance of single-buffered MINs

1) Normalized throughput for single-buffered MINs

Figure 5 represents the normalized throughput of a single-buffered *i*-stage MIN, where i = 3, 4, 6, 8, 10, versus the probability of packet arrivals. In the diagram,

curves Num-L = i and Simu-L = i depict the normalized throughput of a single-buffered i-stage MIN estimated by the analytical model and by a simulation respectively.

From Figure 5 it is obvious that the normalized throughput deteriorates as the network size increases.

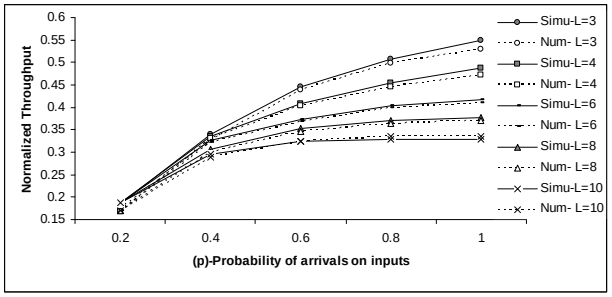

Figure 5 Normalized throughput of an i-stage MIN versus probability of arrivals according to the analytical model and the simulation

It is worth noting that the numerical results of both methods have been found to be in close agreement (differences were less than 1%).

2) Blocking probabilities for single-buffered MINs

Figure 6 illustrates the *blocking probabilities* (P_b) per stage versus the *probability of packet arrivals* (p) at inputs. In the diagram, curves L = X-Num and L = X-Simu depict the *blocking probabilities* (P_b) at layer X, where X = 1, 2, …, 8 of an single-buffered eight-stage MIN estimated by the analytical model and simulation respectively.

According to this diagram, the *blocking probabilities* (P_b) in the first layers are greater, while in the last layer there is no blocking. The numerical results of the two methods have been found again to have the same close agreement. The *blocking probability* decreases with the number of stages. So, the use of an asymmetric buffer size can be proposed. An implementation that is with a buffer size that is larger in the first layer and becomes gradually smaller during the following stages can be used as an optimal cost-effective solution. This technique may also improve the performance of the MINs. So, this configuration can be applied in the design of large scale MINs, in order to develop high-speed networks.

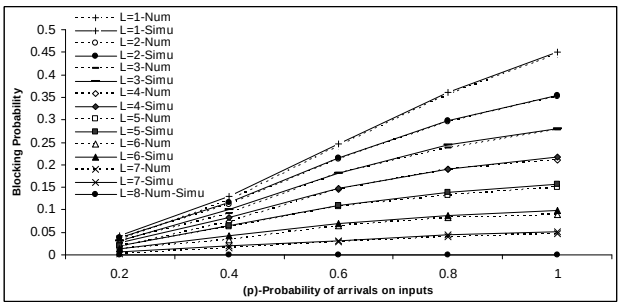

Figure 6. Blocking probabilities (P_b)/stage of a eight-stage (256 × 256) MIN versus probability of arrivals according to the analytical model and simulation

Altering this special analytical method, the calculation of the performance evaluation of the above described asymmetric – with respect to the buffer size – MIN can be achieved more easily. In this case it is only necessary to write the *utilization* equation per stage. Then, putting them in the 'forward' and 'backward' sections of the iterative method easily obtains the steady-state of queues' *utilization*.

3) Normalized packets latency on single-buffered MINs

Similarly, Figure 7 represents the *normalized packet latency* of a single-buffered i-stage MIN, where i = 3, 4, 6, 8, 10, versus the *probability of packet arrivals* according to both analytical model and simulation. It is seen that the *normalized latency* becomes higher as the network size increases.

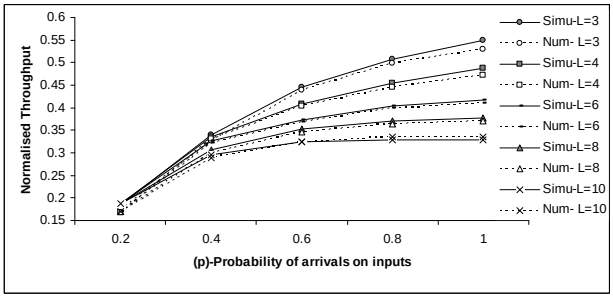

Figure 7. Normalized packets latency of an i-stage MIN versus probability of arrivals according to analytical model and simulation

Low values of packet *latency* are observed for relevant low values of packet arrivals. This happens because the packet population is low in numbers and therefore the number of blocking packets observed is also low. Then, as the offered load rises, the packet *latency* follows this augmentation due to the increment in the backpressure phenomenon. The results obtained by the two methods were again found to be in the same close agreement.

4) Utilization per stage in single-buffered MINs

Finally, Figure 8 presents the *utilization* (u) per stage versus the *probability of arrivals* (p) at inputs for an eight-stage (256 × 256) MIN, where the numerical results obtained by the two methods have again been found to be in the same close agreement. It is worth noting that the *utilization* of the last stage depicts the *throughput* of the MIN because there is no blocking at the last stage.

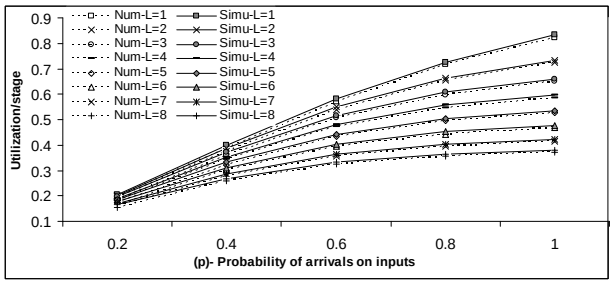

Figure 8. Divergence of utilization/stage versus probability of packet arrivals for an eight-stage MIN

The *throughput* of a MIN is one of the two most significant performance factors – the other is the *latency* – making the multistage fabric suitable for the core and backbone networks which typically provide high capacity

communication facilities. The curves shown in Figures 6 and 8 clearly show that the increment in the offered load provides higher *utilization* and thus *blocking probabilities*. It is also noteworthy that these probabilities have lower values at later stages due to the fact that the last stages are subject to lighter loads, when blocking is heavier.

5) Results for lost and serviced packets' probabilities

The *lost packets at inputs* of the MIN are correlated with the *serviced* packets which have finally been accepted by the system in comparison with the *total* number of packets arriving at inputs. The *probabilities of serviced* packets (or the population of serviced packets) remain constant in each stage, as all the packets lead from the input to the output, because packets cannot be lost in the intermediate stages. Figure 9 presents the *service probabilities* for a single-buffered *i*-stage MIN where *i* = 4, 10 and for variable cases of arriving traffic. As can be seen, the *service probability* remains constant in all stages and that confirms the analysis of Lemma 2, Formula (9). Moreover, the *loss probability* of packets at the MIN's inputs is studied. Thus, Figure 9 illustrates the *lost packets* at inputs of the MIN versus the probability of packets arriving at inputs of an *i*-stage MIN where *i* = 4, 10. As can be seen, the *loss probability* is increased as the arrival rate of packets increases. In the case of low traffic, the values of *lost packets* remain low. On the other hand when traffic is high ($p > 0.7$) the *probability of packets* being lost is over 30%. Furthermore, the equation $p = p_{serv} + p_{lost}$ is confirmed by arithmetic solution, as expected.

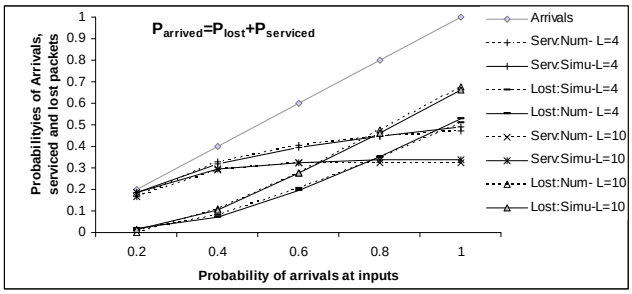

Figure 9. Probabilities of serviced and lost packets for a single-buffered MIN consisting of 2 × 2 SEs

Thus in Figure 9, if we add the values of the curves p_{lost} to the corresponding numbers of the curves $p_{serviced}$, then we obtain the numbers of the $p_{arrived}$ curve, which is an indirect confirmation of our results.

C. Performance of double-buffered MINs

6) Normalized throughput for double-buffered MINs

Figure 10 represents the normalized throughput of a double-buffered i-stage MIN, where i = 3, 4, 6, 8, 10, versus the probability of packet arrivals. In the diagram, curves Num-L = i and Simu-L = i depict the normalized throughput of a double-buffered i-stage MIN estimated by the analytical model and by simulation respectively. From Figure 10 it is obvious that the *normalized throughput* deteriorates as the network size increases.

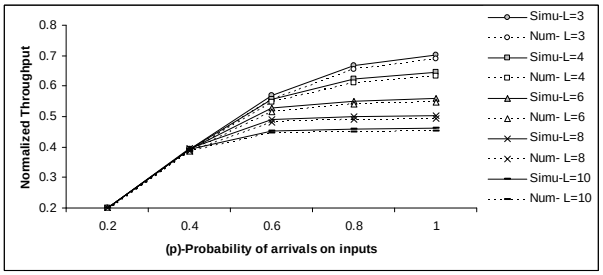

Figure 10 Normalized throughput of an i-stage double-buffered MIN versus probability of arrivals according to the analytical model and simulation

Comparing the values of Figure 10 with the corresponding values of Figure 3, it is obvious that the *throughput* values of double-layer MINs are higher than those of single-buffered MINs with the same configuration set-up.

V. COMBINED PERFORMANCE FACTOR

A. Combined performance factor for multi-criteria evaluation of MINs

In general, performance evaluation factors can be divided into two major sets: factors to be maximized (e.g. *throughput*) and factors to be minimized (e.g. *latency*, *cost*, etc.). Let the first maximized set be $x_{\max} = \{x_{1,\max}, x_{2,\max,}, ..., x_{\mu,\max}\}$ of normalized performance metrics and let the minimized set of normalized performance metrics be $y_{\max} = \{y_{1,\max}, y_{2,\max,}, .., y_{\nu,\max}\}$, where μ, ν are the numbers of factors to be maximized and minimized respectively.

Nevertheless, it is interesting to have a general evaluation using only one factor. This factor must suggest better overall performance, that is, when the first factor's set is maximized and the second factor's set is minimized simultaneously. We call this factor the *Combined Performance Factor* (CPF) and it is given by the following formula:

$$CPF = \sqrt{\sum_{i=1}^{\mu} x_{i,\max}^2 + \sum_{i=1}^{\nu} \left(\frac{1}{y_{i,\min}} \right)^2}$$

In any multi-criteria decision-making problem, however, the importance of each criterion is a design problem. Therefore, when it is of interest to give a weight (concerning the importance in the network) to each separate metric then the above formula can be replaced by:

$$CPF(w_i, w_j) = \sqrt{\frac{\sum_{i=1}^{\mu} w_i \cdot (x_{i,\max})^2 + \sum_{j=1}^{v} w_j \left(\frac{1}{y_{j,\min}}\right)^2}{\sum_{i=1}^{\mu} w_i + \sum_{j=1}^{v} w_j}}$$

where w_i, w_j are the corresponding weights of the normalized system's parameters. According to this equation, when the $x_{i,\max}$ metrics become larger and/or the $y_{i,\min}$ metrics become smaller, the CPF becomes larger. The reference value domain of CPF ranges from 0 to 1.

The main condition which must be satisfied when the CPF factor is applied is the assumption that $y_{j,\min} \neq 0$. Besides this, all the measured factors must be calculated and manipulated as inter-individual metrics.

In this paper, we use the most important performance indicators of *normalized thoughput* (T_{hN}) and *normalized latency* $(\overline{D_N})$. It is obvious that the performance of a MIN is considered optimal when (T_{hN}) is maximized while $\overline{D_N}$ is minimized. Consequently, the formula for computing the CPF acts so that the overall performance metric follows that rule. Formally, CPF can be simplified to:

$$CPF\ (w_{Th}, w_D) = \sqrt{\frac{w_{Th}.T_{hN}^2 + w_D \cdot \left(\frac{1}{\overline{D_N}}\right)^2}{w_{Th} + w_D}},$$

where w_{Th} and w_D denote the corresponding weights of the two performance metrics participating in the overall performance factor CPF, designating its importance for the corporate environment. According to this equation, when the *throughput* metric becomes larger and/or the latency becomes smaller, the CPF becomes larger. The reference value domain of CPF ranges from 0 to 1. Consequently, as the CPF becomes higher, the performance of the MIN is considered to improve.

B. Applying the Combined performance factor

The role of buffer size in MINs

Figures 11 and 12 depict the behavior of the CPF for single- and double-buffered MINs correlated with the *offered load* under various *network sizes*, where different *weights* for each factor participating in the *CPF* are considered, thus designating that factor's importance in the corporate environment; for example, for batch data transfers *throughput* is more important, whereas for streaming media the *latency* must be optimized. According to these figures, solid curves SB-L = *i* represent the overall performance metric *CPF* for single-buffered *i*-stage MINs, while dotted curves DB-L = *i* stand for the corresponding double-buffered configurations where *i* = 4, 6, 8.

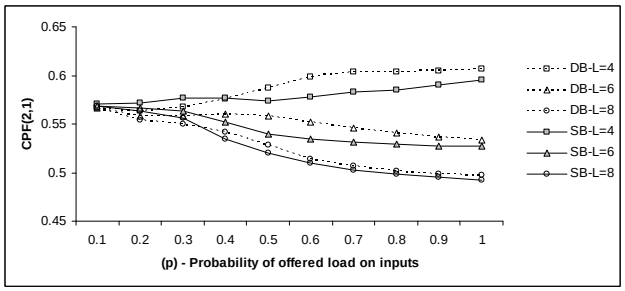

Figure 11. CPF versus (p) probability of packet arrivals at i-stage MINs ($w_{th} = 2$, $w_d = 1$)

In the first diagram the *throughput* factor is considered to be of double importance ($w_{th} = 2$, $w_d = 1$), while in the second diagram the *latency* factor is assumed to be of twofold significance ($w_{th} = 1$, $w_d = 2$).

In Figure 11, where *throughput* is more important, two areas may be identified: the first one spans the 'light input load' segment of the x-axis in which single-buffer configurations offer slightly better overall performance, and the second one spans the 'medium- and high-load' segment of the x-axis in which the gain for the *CPF* metric of double-buffered MINs is considerable.

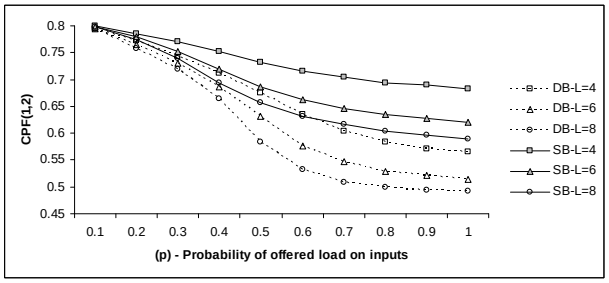

Figure 12. CPF versus (p) probability of packet arrivals at i-stage MINs ($w_{th} = 1$, $w_d = 2$)

On the other hand, when the *latency* is assumed to be of twofold significance (Figure 12), it is seen that all single-buffered set-ups exhibit improved overall performance compared with the corresponding double-buffered ones. Consequently, the findings of this metric can be used by network designers for drawing optimal configurations while setting up MINs to best meet the overall performance and cost requirements under the anticipated traffic load and quality of service specifications, where performance prediction before actual network implementation can also minimize deployment cost and rollout time.

VI. METHODOLOGY'S EXPANDABILITY

This methodology can be extended to deal with performance calculations in modern, more complicated MIN architectures [19, 22] which are directed at new types of applications.

The general idea of performance calculations is as follows. Because MINs have a compound structure, they can be analyzed in stages or modules (usually buffers). Every module can be studied in an arbitrary time cycle of

its operation. From this study an equation is extracted which describes the state and the state transitions in question (e.g. a utilization equation). Afterwards, the equations of the sequence stages (with their relevant boundary conditions) are put into the sections of the iterative method. The convergence of the algorithm gives values of performance indicators that underpin the system when it is in a state of equilibrium.

Some examples which exploit this fast arithmetical convergent method are presented in the following cases:

1) MINs that support traffic with two or more priority classes

In this case the packets entering the fabric are distinct in two or more priority classes. The higher priority classes always earn the memory space in comparison with other packets with lower class priority.

The packet priority processes of packet forwarding can be modeled by parallel queues' pipelines. There are as many parallel processes as there are priority classes. Each parallel process behaves as a single priority model (like the presented model). The current Lemma 2 remains the same for each priority class inasmuch as the packets entering the fabric cannot be lost while they are forwarded to the outputs.

A noteworthy point here is the blocking probabilities at the last stage. In the last stage, the packets with highest class priority do not suffer from blocking. Contrary to this, the packets of a lower priority class may have been blocked owing to the existence of higher priority traffic. The blocking probability of a lower class of priority traffic in the last stage is calculated by the proliferation of queue utilization of each last-stage upper class's priority. This is the relation which connects the packet priority classes. Afterwards, the equations are formed on the basis of the sub-section 4.1 analysis, taking into account the blocking probabilities which appeared in the last stage.

The extracting equations are put within the iterative algorithm's sections. Running the algorithm until it converges provides indicators about the state of equilibrium. This method gives a solution to the issue of a large number of priority classes.

2) MINs supporting multicast traffic

Most important in this case is that in a stage of the MIN the packets increment which appears is caused by the multicasting operation. Thus, the basic condition of Lemma 2 cannot be true because the fabric does not act as a pipeline, since the number of entering packets does not remain the same as they are forwarded from stage to stage. The last stage has a high density of packets and this amount is reduced to that of the preceding stages by a factor which is equal to $\frac{1}{1+w}$, where (w) is the multicast ratio which denotes the multicast packets population divided by the total packet population in a stage. This factor is considered to be fixed for all stages.

Taking into account the packet reduction from the last stage to the first and working in the same way as in Lemma 2 of this paper, we extract a modifying Lemma 2.

Then, in the same way, as shown in Section 4.1, a *utilization* formula can be extracted and in consequence the same iterative algorithm can be applied.

3) MINs with variable buffer sizes among the stages

In this case the *utilization* formulae do not remain the same for all stages. Therefore, following the analysis described above in sub-section 4.1 we can obtain relevant *utilization* formulae for each stage. In consequence, the ('forward' and 'backward') sections of iterative algorithms are compounded stage by stage, putting the stage's *utilization* equations with their boundary conditions. The convergence of the iterative algorithm also gives the equilibrium value of the *utilization* metric.

Finally, Markov processes are often proposed for modeling and evaluating MINs in parallel or distributed systems. Simulation based on Markov chains provides a powerful method for performance evaluation. But it comes with a huge drawback: it often requires long run times until accurate results are determined with high confidence levels.

On the other hand, the adaptation, accuracy, and fast convergence are the main advantages of the method presented above, particularly compared with Markovian analytical approaches.

The exemplification of the current arithmetical method is not limited. This approximate method can be applied even in other cases of modern MIN architectures, making their performance evaluation attainable.

VII. CONCLUSIONS AND FUTURE WORK

Today's gigabit Ethernet and ATM switches, terabit routers, multiprocessor systems, and general parallel systems are typical applications of interconnection networks which have been identified as efficient components in communication structures.

In this paper, a performance methodology for Multistage Interconnection Networks (MINs) is presented. The performance methodology goals were threefold. Firstly, it incorporates an analytical method which gives fast and accurate results based on an iterative algorithm which converges quickly, giving performance metrics as separate factors in the state of equilibrium. Secondly, it is accompanied by a general evaluation factor which helps us in choosing MINs which perform better in comparison with other similar MIN architectural specification and design goals. Thirdly, it presents expandability of several MIN architectural requirements.

The methodology was exemplified for the case of symmetrical MINs comprising 2 × 2 single- or double-buffered SEs. This model represents a real type of blocking (backpressure) which is a very common phenomenon for SEs. This new approximate analytical method verifies the anticipated fact that the *blocking probability* and the *utilization* will get smaller when moving from the first stage to the last one (the last stage has zero *blocking probability*).

The results obtained by a thorough study are confirmed by simulation. It was found that the results of our approximate method are in close agreement (differences are less than 2%) with the corresponding simulation

experiments. Additionally, the results in some cases were validated by existing related work in the literature.

Moreover, the performance factor which is defined for multi-criteria evaluation of MINs can play a significant role in decision-making for MIN selection.

The process and results which are obtained by this performance analysis can be a useful tool for analyzing communication, especially in the area of parallel systems, and moreover it can be a useful tool for designers or teams evaluating, monitoring, or optimizing systems.

The main advantage of the proposed performance evaluation methodology is its flexibility, owing to its ability to be adapted easily to MINs' various architectural requirements and their operations. Therefore, it could be the main platform for testbed and performance analysis in some special modern subjects like MINs supporting traffic with priorities or multicast traffic, or MINs which operate with retransmission packets.

In future work we will consider such cases and will make efforts to provide MIN designers with metrics that will support them in choosing the best MIN configuration, taking into account the applications (e.g. multimedia streaming versus file transfer) that the MIN will support.

APPENDIX A

CONVERGENT ALGORITHM

Let $_{[m]}u^{(i)}$ be the value of $u^{(i)}$ during the m-th iteration of the following algorithm:

Algorithm I

BEGIN
m := 0
/*Start of PHASE A (Initialize Backward Solution)*/
DO
BEGIN
Initialize: $_{[0]}u^{(i)} := p$ /* for stages L,…,1*/
END FOR
/* End of PHASE A */

REPEAT
m:= m + 1
/* Start of PHASE B (Forward Solution) */
Calculate: $_{[m]}u^{(1)}$ (formulae (15), using packets arrivals (p)) /*for stage 1 */
FOR i = 2 TO L-1 DO
BEGIN
Calculate: $_{[m]}u^{(i)}$ (formulae (15)) /*for stages 2,…, *L-1* */
END FOR
Calculate: $_{[m]}u^{(L)}$ (formulae (15), with $p_b^{(i+1)} = 0$)
/*for stage *L* */
/* End of PHASE B (Forward Solution) */
m := m + 1
/* Start of PHASE A (Backward Solution) */
Calculate: $_{[m]}u^{(L)}$ (formulae (15), with $p_b^{(i+1)} = 0$)
/*for stage *L* */
FOR i = L-1 DOWNTO 2 DO
BEGIN
Calculate: $_{[m]}u^{(i)}$ (formulae (15)) /*for stages (*L-1)*,.., 2*/
END FOR
Calculate: $_{[m]}u^{(1)}$ (formulae (15), using packet arrivals (p)) /*for stage 1 */
/* End of PHASE A (Backward Solution) */
UNTIL $(_{[m]}u^{(i)} - _{[m-1]}u^{(i)}) < \varepsilon$ for all stages $i = 1$ to L
Set $u^{(i)}$ to the values of $_{[m]}u^{(i)}$ for all stages $i = 1$ to L
Calculate $p_b^{(i)}$, $\overline{D}$, $\overline{D}_N$

REFERENCES

[1] Ranjan Kumar Dash, Nalini Kanta Barpanda, and Chita Ranjan Tripathy, "A New and Efficient method to Evaluate Residual Broadcast Reliability of Fault-tolerant Multistage Interconnection Networks", *IJCSNS International Journal of Computer Science and Network Security*, VOL.8 No.9, September 2008.

[2] Kumar, S., "Mathematical Modelling and Simulation of a Buffered Fault Tolerant Double Tree Network", *Advanced Computing and Communications*, 2007. ADCOM 2007. International Conference on Volume , Issue , 18-21 Dec. 2007 Page(s):422 – 433

[3] Atiquzzaman M. and M.S. Akhatar, "Efficient of Non-Uniform Traffic on Performance of Unbuffered Multistage Interconnection Networks", *IEE Proceedings Part-E*, 1994.

[4] Bouras C., Garofalakis J., Spirakis P., Triantafillou V., "A general performance model for multistage inteconnection networks", *Euro-Par'97*. August 25-29.

[5] Bouras C., Garofalakis J., Spirakis P., Triantafillou V., "An analytical performance model for multistage interconnection networks with finite, infinite and zero length buffers, in *Performance Evaluation 34*(1998) 169-182.

[6] Bouras C., Garofalakis J., Spirakis P., Triantafillou V., "Queuing delays in differed multistage interconnection networks", *in Proc. 1987 ACM Simetrics Conf.,* May 11-14, 1987, Banff, Alberta, Canada, pp. 111-121.

[7] Y.-C. Jenq, "Performance analysis of a packet switch based on single –buffered banyan networks", *IEEE Journal Selected Areas of Commun. SAS-1(6)* (1983), 1014-1021

[8] Garofalakis J., P. Spirakis, "The performance of multistage interconnections networks with finite buffers", in :*Proc. ACM SIGMETRICS Conf.,* 1990, short paper.

[9] Cisco Systems, http://newsroom.cisco.com/dlls/2004/next_generation_networks_and_the_cisco_carrier_routing_system_overview.pdf .

[10] Hsiao S.H. and Chen R. Y., "Performance Analysis of Single-Buffered Multistage Interconnection Networks", *3rd IEEE Symposium on Parallel and Distributed Processing*, pp. 864-867, December 1-5, 1991.

[11] I-Pyen Lyen, David M. Koppelman, "An Analysis of Banyan Networks Offered Traffic with Geometrically Distributed Message Lengths", *IEE Proceedings – Communications Volume 142, Issue 5*, October 1995 p. 285-291.

[12] Sandeep Sharma, P.K.Bansal, Karanjit Singh Kahlon "On a class of multistage interconnection network in parallel processing", *IJCSNS International Journal of Computer Science and Network Security*, *VOL.8 No.5*, May 2008

[13] Kruskal C.P., Sinir M., The performance of multistage interconnection networks for multiprocessors, *IEEE Trans. Comput. C-32* (1983) 1091-1098.
[14] G. Zheng, T.Wilmarth, P. Jagadishprasad, and L. V. Kale. "Simulation-based performance rediction for large parallel machines", In *International Journal of Parallel Programming*, number to appear, 2005
[15] Lin T., Kleinrock L., "Performance Analysis of Finite-Buffered Multistage Interconnection Networks with a General Traffic Pattern", *Joint International Conference on Measurement and Modeling of Computer Systems*, *Proceedings of the 1991 ACM SIGMETRICS conference on Measurement and modeling of computer systems*, San Diego, California, United States, Pages: 68 - 78, 1991.
[16] H. Mun and H.Y. Youn, "Performance analysis of finite buffered multistage interconnection networks", *IEEE Trans. Comput. 43(2)* (1994), 153-161.
[17] D.C. Vasiliadis, G.E. Rizos, C. Vassilakis "Performance Analysis of blocking Banyan Switches", *Proceedings of the IEEE sponsored CISSE 06*, December, 2006.
[18] John Garofalakis, El. Stergiou "An analytical performance model for multistage interconnection networks with blocking" *Sixth Annual Conference on Communication Networks and Services Research* (CNSR2008) Halifax, Nova Scotia, Canada. May 5 - 8, 2008.
[19] Sandeep Sharma, K.S. Kahlon, P.K. Bansal and Kawaljeet Singh, "Irregular Class of Multistage Interconnection Network in Parallel Processing", *Journal of Computer Science,* 01-MAR-2008
[20] Theimer T.H., Rathgeb E. P., and Huber M.N., "Performance Analysis of Buffered Banyan Networks", *IEEE Transactions on Communications,* vol. 39, no. 2, pp. 269-277, February 1991.
[21] D. Tutsch, M. Brenner. "A Multistage Interconnection Network Simulator". *In 17th European Simulation Multiconference: Foundations for Successful Modelling & Simulation (ESM'03);* Nottingham, SCS, pp. 211. 216, 2003.
[22] D. Tutsch, G. Hommel. "Generating Intrconnection Network Simulator. Generating Systems of Equations for Performance Evaluation of Buffered Multistage Interconnection Networks". *Journal of Parallel and Distributed Computing,* 62, no. 2: pp. 228..240,2002
[23] Bianco Andrea, Finochietto Jorge, Mellia Marco, and Neri Fabio, "Multistage Switching Architectures for Software Routers". *IEEE Network* –July/August 2007.
[24] Aydin O. Balkan, Gang Qu, Uzi Vishkin, "An Area-Efficient High-Throughput Hybrid Interconnection Network for Single-Chip Parallel Processing", *Design Automation Conference,* 2008. DAC 2008, 45th *A*CM/IEEE Publication Date: 8-13 June 2008
[25] Raja J., S. Shanmugavel, "Performance Studies of Banyan ATM Switching Networks using RS Codes", *IE Journal-CP,* Vol 84, May 2003..

John Garofalakis (http://athos.cti.gr/garofalakis/index_en.htm) is Associate Professor at the Department of Computer Engineering and Informatics, University of Patras, Greece, and Director of the applied research department "Telematics Center", of the **R**esearch **A**cademic **C**omputer **T**echnology **I**nstitute (**RACTI**).

He is responsible and scientific coordinator of several recent European and national IT and Telematics Projects (ICT, INTERREG, etc.).
His publications include more than 100 articles in refereed International Journals and Conferences. His research interests include Web and Mobile Technologies, Performance Analysis of Computer Systems, Computer Networks and Telematics, Distributed Computer Systems, Queuing Theory.

Eleftherios Stergiou is lecturer in the department of *Information Technology and Telecommunications*, at *Epirus Institute of Technology* in Greece since 2000. He is also a research fellow at the University of Patras. He received the B.S. degree in electrical engineering from NTUA, Athens Greece, and he finished his postgraduate studies at the computer science department of the University of Sheffield (1998).
His research interests on performance evaluation of networks integrate by publishing papers in international journals.
Among these interests, computing analytical methods, interconnection networks, parallel and distributed systems, high-speed networks, are included. Mr E. Stergiou is member of IEEE Computer Society.