Modelling and performance study of finite-buffered blocking Multistage Interconnection Networks supporting natively 2-class priority routing traffic

D. C. Vasiliadis[a,b,*] G. E. Rizos[a,b] C. Vassilakis[a]

[a] Department of Computer Science and Technology, University of Peloponnese, GR-221 00, Tripolis, Greece

[b] Network Operations Center, Technological Educational Institute of Epirus, GR-471 00, Arta, Greece

Abstract

In this paper, we model, analyze and evaluate the performance of a 2-class priority architecture for finite-buffered Multistage Interconnection Networks (MINs). The MIN operation modelling is based on a state diagram, which includes the possible MIN states, transitions and conditions under which each transition occurs. Equations expressing state and transition probabilities are subsequently given, providing a formal model for evaluating the MIN's performance. The proposed architecture's performance is subsequently analyzed using simulations; operational parameters, including buffer length, MIN size, offered load and ratios of high priority packets which are varied across experiments to gain insight on how each parameter affects the overall MIN performance. The 2-class priority MIN performance is compared against the performance of single priority MINs, detailing the performance gains and losses for packets of different priorities. Performance is assessed by means of the two most commonly used factors, namely packet throughput and packet delay, while a performance indicator combining both individual factors is introduced, computed and discussed. The findings of this study can be used by network and interconnection system designers in order to deliver efficient systems while minimizing the overall cost. The performance evaluation model can also be applied to other network types, providing the necessary data for network designers to select optimal values for network operation parameters.

Keywords: Multistage Interconnection Networks, Banyan Switches, Packet Switching, Quality of Service, Priority, Performance Analysis, Simulation model

1 Introduction

Multistage Interconnection Networks (MINs) with crossbar Switching Elements (SEs) are often used in the context of multiprocessor computer architecture for the interconnection of processors and memory modules [9], [34] MINs are also increasingly adopted for implementing the switching fabric of high-capacity communication processors, including ATM switches, gigabit Ethernet switches and terabit routers [2], [13], [29], [30] [41]. MINs owe their popularity both to operational features they deliver, such as the ability to route multiple communication tasks concurrently, and to the low cost/performance ratio they achieve. The family of multistage interconnection networks includes several major categories, such as Omega, Generalized Cube, Benes, Batcher Banyan etc; of these, MINs with the Banyan [17] property are more widely adopted, since non-Banyan MINs are, in general, more expensive than Banyan ones and more complex to control.

The performance of the communication infrastructure that interconnects the system's elements (nodes, processors, memory modules etc) has been recognized as a critical factor for overall system performance, both in the context of parallel and in the context of distributed systems. As a result, much research has been conducted, targeting to identify the factors that affect the communication infrastructure's performance and provide models for performance prediction and evaluation. Two major directions have been taken to this end: the first employs analytical models based either on Markov models or on Petri-nets, while the second uses simulation techniques. These works enable network designers to estimate network performance before it is actually implemented, allowing thus network design tuning and adjustment of parameters. Using the insights from this procedure, network designers may craft efficient systems, tailored to the specific requirements of the system under implementation with a minimal cost, since the actual implementation decisions are deferred until all operational parameters have been determined.

In this paper we propose a novel approach to model the operational behaviour of a 2-priority class MIN, which takes into account the previous and the current state of both queues (high and low priority) of each switching element, leading thus to more accurate results. The modelling scheme is complemented with equations expressing the probability for each state transition, giving a complete analytic framework for 2-class priority MIN performance behaviour. Simulation experiments are also conducted to estimate the MIN performance under various traffic loads, buffer lengths, high/low priority traffic ratios and MIN sizes (number of stages). The findings of this paper can be used by network designers to gain insight on the impact of each MIN design parameter on the overall MIN performance and to select the optimal MIN configuration for the needs of their environment.

The remainder of this paper is organized as follows: section 2 overviews related work in the area of network performance evaluation and priority schemes, while in section 3 we present the proposed 2-class priority scheme, we describe its operation and give its analytical system of equations. Thus, a novel 5-state and 6-state buffer model for high and low priority queues respectively is employed. Subsequently, in section 4 we present the performance criteria and parameters related to the network. Section 5 presents and discusses the results of our performance analysis, which has been conducted through simulation experiments, while section 6 provides the concluding remarks and outlines future work.

2 Related work

Single priority queuing systems in the context of MIN performance evaluation have been extensively studied and are reported in numerous publications. For example, [15], [21], [23], [33], [39] study the *throughput* and system *delay* of a MIN assuming the SEs have a single input buffer, whereas the performance of a finite-buffered MINs is studied, among others, in [16], [22], [31]. In the industry domain, Cisco has used multistage switch fabric for building some of its new routers, such as the CRS-1 router [3], [4]. The switching fabric providing the communications path between line cards is 3-stage, self-routed architecture.

A recent development in the MIN domain is the introduction of dual priority (or 2-class) queuing systems [38], [39], which are able to offer different quality-of-service parameters to packets that have different priorities. Packet priority has been a common issue in networks, arising when some packets need to be offered better quality of service than others. Packets with real-time requirements (e.g. from streaming media) vs. non real-time packets (e.g. file transfer), and out-of-band data vs. ordinary TCP traffic [32] are two examples of such differentiations. Cases of different priorities may arise also in the context of parallel architectures, e.g. some CPUs may be running operating system processes and traffic between these CPUs and memory modules can be prioritized against traffic from/to other CPUs. In all these cases, the communications infrastructure should include provisions to (a) allow applications or architectural components to designate packet priority and (b) offer better quality of service to the packets indicated as "high priority" ones.

While the 802.1D standard [26] specifies eight priority levels and the Diffserv standard [14] specifies six "class selectors", it has been anticipated that few switches will actually provide support for eight priority classes [19], and hence IEEE 802.1Q provides recommended mappings from the eight priority classes specified in 802.1D to fewer queues [25]. Many contemporary switches prioritize packets through a process involving the steps of *classification*, *marking* and *queuing* (with a *policing* step also appearing in some cases) [5], [18], and the outcome of this process is the placement of the packets in two queues in the typical case (e.g. [1], [10], [11], [20], [24], [40]), with a higher number of queues being supported by higher-end switches (e.g. four in [5] , [12], [18] and eight in [6], [7]). Thus, in this paper we focus on studying MINs that natively support two priority levels.

There are already several commercial switches which accommodate traffic priority schemes, such as [1], [10], [11], [20], [24], [40]. These switches consist internally of single priority SEs and employ two priority queues for each input port, where packets are queued based on their priority level. Chen and Guerin [8] studied an (N x N) non-blocking packet switch with input queues, built using one-priority SEs. Ng and Dewar [27] introduced a simple modification to load-sharing replicated buffered Banyan networks to guarantee priority traffic transmission.

3 Modelling and analytical equations for a 2-class packet priority MIN

A Multistage Interconnection Network (MIN) can be defined as a network used to interconnect a group of N inputs to a group of M outputs using several stages of small size Switching Elements (SEs) followed (or preceded) by link states. Its main characteristics are its topology, routing algorithm, switching strategy and flow control mechanism. A MIN with the Banyan property is defined in [17] and is characterized by the fact that there is exactly a unique path from each source (input) to each sink (output). Banyan MINs are multistage self-routing switching fabrics. Thus, each SE of k^{th} stage, where k=1...n can decide in which output port to route a packet, depending on the corresponding k^{th} bit of the destination address.

An (N X N) MIN can be constructed by n=$\log_c N$ stages of (cxc) SEs, where c is the degree of the SEs. A typical SE is illustrated in fig. 1. At each stage there are exactly N/c SEs, consequently the total number of SEs of a MIN is $(N/c)*\log_c N$. Thus, there are O(N*log$N$) interconnections among all stages, as opposed to the crossbar network which requires O(N^2) links.

In a 2-class priority scheme, when a packet is entered in the MIN its priority is specified by the application or the architectural module that has produced the packet. The priority is henceforth reflected into a bit in the packet header and is maintained throughout the lifetime of the packet within the MIN.

In order to support priority handling, each SE has two transmission queues per link, accommodated in two (logical) buffers, with one queue dedicated to high priority packets and the other dedicated to low priority ones. During a single network cycle, the SE considers all its links, examining for each one of them firstly the high priority queue. If this is not empty, it transmits the first packet towards the next MIN stage; the low priority queue is checked only if the corresponding high priority queue is empty. Packets in all queues are transmitted in a first come, first served basis. In all cases, at most one packet per link (upper or lower) of an SE will be forwarded for each pair of high and low priority queues to the next stage.

A typical configuration of a 3-stage MIN consisting of 2x2 SEs is depicted in fig. 1. This configuration is based on the standard 8x8 delta network setup proposed by Patel [28], but has been extended to use queues per input link (for high and low priority packets).

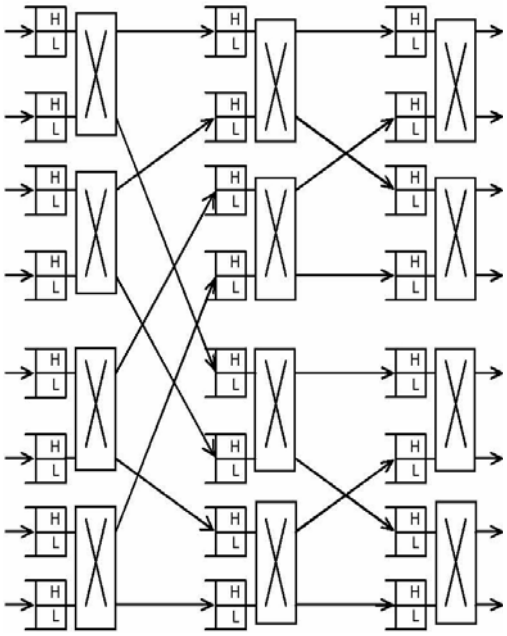

Fig.1. 2-class priority for 3-stage MIN consisting of 2x2 SEs

In this paper, we consider a Multistage Interconnection Network with the Banyan property that operates under the following assumptions:

- 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 one. In the second phase, packets flow from one stage to the next in accordance to the flow control information.
- The arrival process of each input of the network is a simple Bernoulli process, i.e. the probability that a packet arrives within a clock cycle is constant and the arrivals are independent of each other. We will denote this probability as λ. This probability can be further broken down to λ_h and λ_l, which represent the arrival probability for high and low priority packets, respectively. It holds that $\lambda = \lambda_h + \lambda_l$.
- Under the two-class priority mechanism, when applications or architectural modules enter a packet to the network they specify its priority, designating it either as high or low. The criteria for priority selection may stem from the nature of packet data [32] (e.g. packets containing streaming media data can be designated as

high-priority while FTP data can be characterized as low-priority), from protocol intrinsics (e.g. TCP out-of-band/expedited data vs. normal connection data) or from properties of the interconnected system architecture elements.

- A high/low priority packet arriving at the first stage (k=1) is discarded if the high/low priority buffer of the corresponding SE is full, respectively.
- A high/low priority packet is blocked at a stage if the destination high/low priority buffer at the next stage is full, respectively.
- Both high and low priority packets are uniformly distributed across all destinations, and each high/low priority queue uses a FIFO policy for all output ports.
- When two packets at a stage contend for a buffer at the next stage and there is no adequate free space for both of them to be stored (i.e. only one buffer position is available at the next stage), there is a conflict. Conflict resolution in a single-priority mechanism operates under the following scheme: one packet will be accepted at random and the other will be blocked by means of upstream control signals. Under the 2-class priority scheme, the conflict resolution procedure takes into account the packet priority: if one of the received packets is a high-priority one and the other is a low priority packet, the high-priority packet will be maintained and the low-priority one will be blocked by means of upstream control signals; if both packets have the same priority, one packet is chosen randomly to be stored in the buffer whereas the other packet is blocked. The priority of each packet is indicated through a priority bit in the packet header, thus it suffices for the SE to read the header in order to make a decision on which packet to store and which to drop.
- All SEs have deterministic service time.
- Finally, all packets in input ports contain both the data to be transferred and the routing tag. In order to achieve synchronously operating SEs, the MIN is internally clocked. As soon as packets reach a destination port they are removed from the MIN, so, packets cannot be blocked at the last stage.

Our analysis introduces a model, which considers not only the current state of the associated buffer, but also the previous one, i.e. in the case of a single-buffered MIN based on the one clock history consideration we

enhance the Mun's [23] three states model with a five states buffer model, which is described in the following paragraphs.

3.1 State notations for high priority queues

- *State '00^h'*: High priority buffer was empty at the beginning of the previous clock cycle and it is also empty at beginning of the current clock cycle (i.e. no new high priority packet has been received during the previous clock cycle; high priority buffer remains empty).
- *State '01^h'*: High priority buffer was empty at the beginning of the previous clock cycle, while it contains a new high priority packet at the current clock cycle (i.e. a new high priority packet has been received during the previous clock cycle; high priority buffer is filled now).
- *State '10^h'*: High priority buffer had a high priority packet at the previous clock cycle, while it contains no packet at the current clock cycle (i.e. a high priority packet has been sent during the previous clock cycle, but no new such packet has been received; high priority buffer is empty now).
- *State '11n^h'*: High priority buffer had a high priority packet at the previous clock cycle and has a new one at the current clock cycle (i.e. a high priority packet has been sent during the previous clock cycle, and a new such packet has also been received; high priority buffer is filled with a new high priority packet now).
- *State '11b^h'*: High priority buffer had a high priority packet at the previous clock cycle and has the packet blocked at the current clock cycle (i.e. no high priority packet has been sent during the previous clock cycle due to blocking; high priority buffer is filled with a blocked high priority packet now).

3.2 Definitions for high priority queues

The following variables are defined in order to develop an analytical system of equations. In all definitions SE(k) denotes a SE at *stage k* of the MIN.

- $P_{00}(k,t)^h$ is the probability that a high priority buffer of SE(k) is empty at both $(t\text{-}1)^{th}$ and t^{th} network cycles.

- $P_{01}(k,t)^h$ is the probability that a high priority buffer of SE(k) is empty at (t-1)th network cycle and has a new packet at t^{th} network cycle.
- $P_{10}(k,t)^h$ is the probability that a high priority buffer of SE(k) has a packet at (t-1)th network cycle and is empty at t^{th} network cycle.
- $P_{11n}(k,t)^h$ is the probability that a high priority buffer of SE(k) has a packet at (t-1)th network cycle and has also a new one at t^{th} network cycle.
- $P_{11b}(k,t)^h$ is the probability that a high priority buffer of SE(k) has a packet at (t-1)th network cycle and has a blocked one at t^{th} network cycle.
- $q(k,t)^h$ is the probability that a high priority packet is ready to be sent to a high priority buffer of SE(k) at t^{th} network cycle (i.e. a high-priority packet will be transmitted by an SE(k-1) to SE(k)).
- $r_{01}(k,t)^h$ is the probability that a high priority packet in a buffer of SE(k) is ready to move forward during the t^{th} network cycle, given that the buffer is in '01^h'state.
- $r_{11n}(k,t)^h$ is the probability that a high priority packet in a buffer of SE(k) is ready to move forward during the t^{th} network cycle, given that the buffer is in '$11n^h$' state.
- $r_{11b}(k,t)^h$ is the probability that a high priority packet in a buffer of SE(k) is ready to move forward during the t^{th} network cycle, given that the buffer is in '$11b^h$' state.

3.3 Mathematical analysis for high priority queues

The following equations, which are derived from the state transition diagram at fig. 2, represent the state transition probabilities as clock cycles advance.

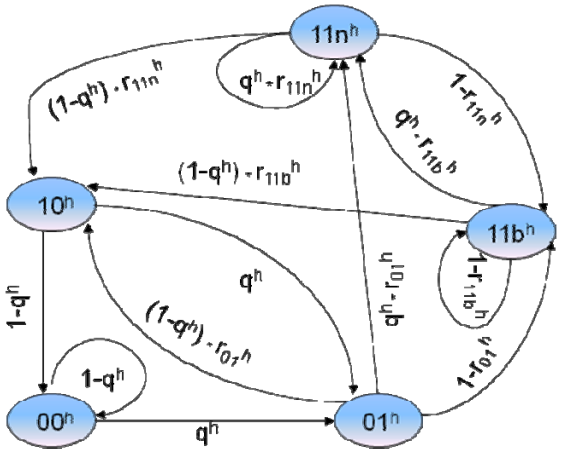

Fig.2. A state transition diagram for a high priority buffer of SE(k)

The probability that a high priority buffer of SE(k) was empty at the (t-1)th network cycle is $P_{00}(k,t\text{-}1)^h$ + $P_{10}(k,t\text{-}1)^h$. Therefore, the probability that a high priority buffer of SE(k) is empty both at the current t^{th} and previous (t-1)th network cycles is the probability that the SE(k) was empty at the previous (t-1)th network cycle multiplied by the probability [1-q(k,t-1)^h] of no high priority packet was ready to be forwarded to SE(k) during the previous network cycle (the two facts are statistically independent, thus the probability that both are true is equal to the product of the individual probabilities). Formally, this probability $P_{00}(k,t)^h$ can be expressed by

$$P_{00}(k,t)^h = [1\text{-}q(k,t\text{-}1)^h] * [P_{00}(k,t\text{-}1)^h + P_{10}(k,t\text{-}1)^h] \quad (1)$$

The probability that a high priority buffer of SE(k) was empty at the (t-1)th network cycle and a new high priority packet has arrived at the current t^{th} network cycle is the probability that the SE(k) was empty at the (t-1)th network cycle [which is equal to $P_{00}(k,t\text{-}1)^h$ + $P_{10}(k,t\text{-}1)^h$] multiplied by the probability $q(k,t\text{-}1)^h$ that a new high priority packet was ready to be transmitted to SE(k) during the (t-1)th network cycle. Formally, this probability $P_{01}(k,t)^h$ can be expressed by

$$P_{01}(k,t)^h = q(k,t\text{-}1)^h * [P_{00}(k,t\text{-}1)^h + P_{10}(k,t\text{-}1)^h] \quad (2)$$

The case that a high priority buffer of SE(k) was full at the (t-1)th network cycle but is empty during the (t-1)th network cycle effectively requires the following two facts to be true: (a) a high priority buffer of SE(k) was full at the (t-1)th network cycle and the high priority packet was successfully transmitted and (b) no high priority packet was received during the (t-1)th network cycle to replace the transmitted high priority packet into the buffer. The probability for fact (a) is equal to $[r_{01}(k,t\text{-}1)^h * P_{01}(k,t\text{-}1)^h + r_{11n}(k,t\text{-}1)^h * P_{11n}(k,t\text{-}1)^h + r_{11b}(k,t\text{-}1)^h * P_{11b}(k,t\text{-}1)^h]$; this is computed by considering all cases that during the network cycle t-1 the SE had a high priority packet in its buffer and multiplying the probability of each state by the corresponding probability that the packet was successfully transmitted. The probability of fact (b), i.e. that no high priority packet was ready to be transmitted to SE(k) during the previous network cycle is equal to $[1\text{-}q(k,t\text{-}1)^h]$. Formally, the probability $P_{10}(k,t)^h$ can be computed by the following formula:

$$P_{10}(k,t)^h = [1\text{-}q(k,t\text{-}1)^h] * [r_{01}(k,t\text{-}1)^h * P_{01}(k,t\text{-}1)^h + r_{11n}(k,t\text{-}1)^h * P_{11n}(k,t\text{-}1)^h + r_{11b}(k,t\text{-}1)^h * P_{11b}(k,t\text{-}1)^h] \quad (3)$$

The probability that a high priority buffer of SE(k) had a packet at the (t-1)th network cycle and has also a new one (different than the previous; the case of having the same packet in the buffer is addressed in the next paragraph) at the t^{th} network cycle is the probability of having a ready high priority packet to move forward at the previous (t-1)th network cycle [which is equal to $r_{01}(k,t\text{-}1)^h * P_{01}(k,t\text{-}1)^h + r_{11n}(k,t\text{-}1)^h * P_{11n}(k,t\text{-}1)^h + r_{11b}(k,t\text{-}1)^h * P_{11b}(k,t\text{-}1)^h$] multiplied by $q(k,t\text{-}1)^h$, i.e. the probability that a high priority packet was ready to be transmitted to SE(k) during the previous network cycle. Formally, this probability $P_{11n}(k,t)^h$ can be expressed by

$$P_{11n}(k,t)^h = q(k,t\text{-}1)^h * [r_{01}(k,t\text{-}1)^h * P_{01}(k,t\text{-}1)^h + r_{11n}(k,t\text{-}1)^h * P_{11n}(k,t\text{-}1)^h + r_{11b}(k,t\text{-}1)^h * P_{11b}(k,t\text{-}1)^h] \quad (4)$$

The final case that should be considered is when a high priority buffer of SE(k) had a high priority packet at the (t-1)th network cycle and still contains the same packet at the t^{th} network cycle. This occurs when the packet in the high priority buffer of SE(k) was ready to move forward at the (t-1)th network cycle, but it was blocked (not forwarded) during that cycle, due to a blocking event - either (a) the associated high priority buffer of the next stage SE was already full due to another blocking, or (b) buffer space was available at stage k+1 but it was

occupied by a second packet of the current stage contending for the same high priority buffer during the process of forwarding. The probability for this case can be formally defined as

$$P_{11b}(k,t)^h = [1-r_{01}(k,t-1)^h] * P_{01}(k,t-1)^h + [1-r_{11n}(k,t-1)^h] * P_{11n}(k,t-1)^h + [1-r_{11b}(k,t-1)^h] * P_{11b}(k,t-1)^h \quad (5)$$

Adding the equations (1) … (5), both left and right-hand sides are equal to 1 validating thus that all possible cases have been covered; indeed, $P_{00}(k,t)^h + P_{01}(k,t)^h + P_{10}(k,t)^h + P_{11n}(k,t)^h + P_{11b}(k,t)^h = 1$ and $P_{00}(k,t-1)^h + P_{01}(k,t-1)^h + P_{10}(k,t-1)^h + P_{11n}(k,t-1)^h + P_{11b}(k,t-1)^h = 1$.

Finally, in the marginal case, when $\lambda_l=0$ (or, equivalently, $\lambda_h=\lambda$), the system of equations (1) … (5) effectively degenerate to the equation system for a single priority MIN.

3.4 State notations for low priority queues

Modelling of low priority queues needs one additional state, as compared to the high-priority queue model, to accommodate the cases that a low priority packet is blocked due to the existence of a high-priority packet in the same link; thus the model for low queues includes six distinct buffer states as follows:

- *State '00^l'*: Low priority buffer was empty at the beginning of the previous clock cycle and it is also empty at beginning of the current clock cycle.
- *State '01^l'*: Low priority buffer was empty at the beginning of the previous clock cycle, while it contains a new low priority packet at the current clock cycle.
- *State '10^l'*: Low priority buffer had a low priority packet at the previous clock cycle, while it contains no packet at the current clock cycle.
- *State '11n^l'*: Low priority buffer had a low priority packet at the previous clock cycle and has a new one at the current clock cycle.
- *State '11b^l'*: Low priority buffer had a low priority packet at the previous clock cycle and has the packet blocked at the current clock cycle.

- *State '11w^{l}'*: Low priority buffer had a low priority packet at the previous clock cycle and has this packet waiting at the current clock cycle, because the corresponding high priority queue has a ready packet to be transmitted; recall that high priority packets have precedence over low priority ones at the transmission process.

3.5 Definitions for low priority queues

Similarly to variable definitions for high priority queues presented in section 3.2, we define here the necessary variables to develop an analytical system of equations for low-priority queues:

- $P_{00}(k,t)^{l}$ is the probability that a low priority buffer of SE(k) is empty at both (t-1)th and t^{th} network cycles.
- $P_{01}(k,t)^{l}$ is the probability that a low priority buffer of SE(k) is empty at (t-1)th network cycle and has a new low priority packet at t^{th} network cycle.
- $P_{10}(k,t)^{l}$ is the probability that a low priority buffer of SE(k) has a low priority packet at (t-1)th network cycle and is empty at t^{th} network cycle.
- $P_{11n}(k,t)^{l}$ is the probability that a low priority buffer of SE(k) has a packet at (t-1)th network cycle and has also a new one at t^{th} network cycle.
- $P_{11b}(k,t)^{l}$ is the probability that a low priority buffer of SE(k) has a packet at (t-1)th network cycle and still has the same packet at t^{th} network cycle, as the packet could not be transmitted due to blocking.
- $P_{11w}(k,t)^{l}$ is the probability that a low priority buffer of SE(k) has a packet at (t-1)th network cycle and still has the same packet at t^{th} network cycle, as the packet could not be transmitted due to the existence of a high priority packet in the same link.
- $q(k,t)^{l}$ is the probability that a low priority packet is ready to be sent to a low priority buffer of SE(k) at t^{th} network cycle (i.e. a low-priority packet will be transmitted by an SE(k-1) to SE(k).
- $r_{01}(k,t)^{l}$ is the probability that a low priority packet in a buffer of SE(k) is ready to move forward during the t^{th} network cycle, given that the buffer is in '01^{l}'state.

- $r_{11n}(k,t)^l$ is the probability that a low priority packet in a buffer of SE(k) is ready to move forward during the t^{th} network cycle, given that the buffer is in '$11n^l$' state.
- $r_{11b}(k,t)^l$ is the probability that a low priority packet in a buffer of SE(k) is ready to move forward during the t^{th} network cycle, given that the buffer is in '$11b^l$' state.
- $r_{11w}(k,t)^l$ is the probability that a low priority packet in a buffer of SE(k) is ready to move forward during the t^{th} network cycle, given that the buffer is in '$11w^l$' state.

3.6 Mathematical analysis for low priority queues

Similarly to subsection 3.3, the following equations, derived from the state transition diagram in fig. 3, represent the state transition probabilities of low priority queues as clock cycles advance.

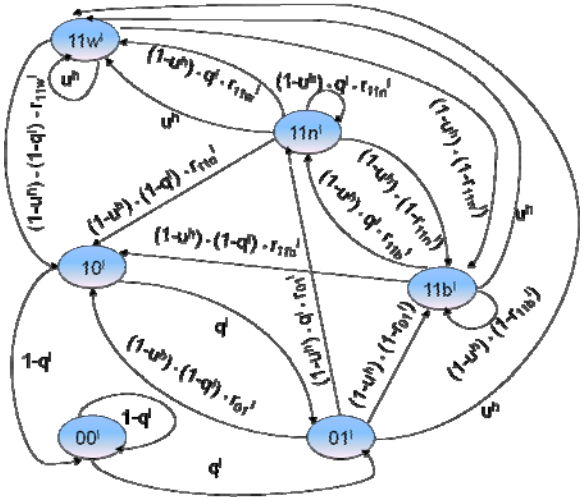

Fig.3. A state transition diagram of a low priority buffer of SE(k)

State probabilities for low priority queues can be formally defined as:

$$P_{00}(k,t)^l = [1-q(k,t-1)^l] * [P_{00}(k,t-1)^l + P_{10}(k,t-1)^l] \quad (6)$$

$$P_{01}(k,t)^l = q(k,t-1)^l * [P_{00}(k,t-1)^l + P_{10}(k,t-1)^l] \quad (7)$$

$$P_{10}(k,t)^l = [1-U(k,t-1)^h] * [1-q(k,t-1)^l] * [r_{01}(k,t-1)^l * P_{01}(k,t-1)^l + r_{11n}(k,t-1)^l * P_{11n}(k,t-1)^l + r_{11b}(k,t-1)^l * P_{11b}(k,t-1)^l + r_{11w}(k,t-1)^l * P_{11w}(k,t-1)^l] \quad (8)$$

$$P_{11n}(k,t)^l = [1-U(k,t-1)^h] * q(k,t-1)^l * [r_{01}(k,t-1)^l * P_{01}(k,t-1)^l + r_{11n}(k,t-1)^l * P_{11n}(k,t-1)^l + r_{11b}(k,t-1)^l * P_{11b}(k,t-1)^l + r_{11w}(k,t-1)^l * P_{11w}(k,t-1)^l] \quad (9)$$

$$P_{11b}(k,t)^l = [1-U(k,t-1)^h] * \{[1-r_{01}(k,t-1)^l] * P_{01}(k,t-1)^l + [1-r_{11n}(k,t-1)^l] * P_{11n}(k,t-1)^l + [1-r_{11b}(k,t-1)^l] * P_{11b}(k,t-1)^l + [1-r_{11w}(k,t-1)^l] * P_{11w}(k,t-1)^l\} \quad (10)$$

$$P_{11w}(k,t)^l = U(k,t-1)^h * [P_{01}(k,t-1)^l + P_{11n}(k,t-1)^l + P_{11b}(k,t-1)^l + P_{11w}(k,t-1)^l] \quad (11)$$

where $U(k,t-1)^h$ expresses the probability that a packet exists in the high priority queue of SE(k) during network cycle t-1 and is given by the following equation:

$$U(k,t-1)^h = r_{01}(k,t-1)^h * P_{01}(k,t-1)^h + r_{11n}(k,t-1)^h * P_{11n}(k,t-1)^h + r_{11b}(k,t-1)^h * P_{11b}(k,t-1)^h \quad (12)$$

The factor $[1-U(k,t-1)^h]$ appearing in the equation effectively manifests that the corresponding states may only be reached if the involved high-priority queues are empty: this holds because the pertinent states may be reached only a packet is transmitted from a low priority queue, and an empty corresponding high-priority queue is a prerequisite for such a transmission to occur.

Adding the equations (6) … (11), both left and right-hand sides are equal to 1, validating thus that all possible cases are covered.; indeed $P_{00}(k,t)^l + P_{01}(k,t)^l + P_{10}(k,t)^l + P_{11n}(k,t)^l + P_{11b}(k,t)^l + P_{11w}(k,t)^l = 1$ and $P_{00}(k,t-1)^l + P_{01}(k,t-1)^l + P_{10}(k,t-1)^l + P_{11n}(k,t-1)^l + P_{11b}(k,t-1)^l + P_{11w}(k,t-1)^l = 1$.

Moreover, in the marginal case, when $\lambda_h=0$ (or, equivalently, $\lambda_l = \lambda$), $U(k,t-1)^h=0$ and thus $P_{11w}(k,t)^l=0$. Consequently, in that case, the system of equations (6) … (10) is equivalent to the system of equations (1) … (5), which is identical to the equation set holding for a single-priority MIN.

The systems of equations which were presented in the previous paragraphs extend the ones presented in other works (e.g. [32]) by considering the state and transitions occurring within an additional clock cycle. This enhancement improves the accuracy of the performance parameters calculation (*throughput* and *delay*). The dependencies among the queues of each SE(k) of the MIN and state transitions presented above have been

incorporated in the simulation logic of the experiments presented in section 5. In our future work, we aim to study in detail an analytical model, incorporating a multi priority scheme and validating the analytical model through simulations. Part of this future work is the analytic computation of the probabilities listed in the definitions section above; currently, all these probabilities are computed through simulation.

4 Performance evaluation methodology

In order to evaluate the performance of a (*N X N*) MIN with $n=\log_c N$ intermediate stages of (*c*x*c*) SEs, we have employed discrete time simulation. In the following text, *T* denotes a relatively large time period divided into *u* discrete time intervals ($\tau_1, \tau_2 \ldots \tau_u$). Performance metrics under the discrete time model may be defined as follows:

- *Average throughput* Th_{avg} is the average number of packets accepted by all destinations per network cycle. This metric is also referred to as *bandwidth*. Formally, Th_{avg} can be defined as

$$Th_{avg} = \lim_{u \to \infty} \frac{\sum_{i=1}^{u} n(i)}{u} \tag{13}$$

  where *n(i)* denotes the number of packets that reach their destinations during the i^{th} time interval.

- *Normalized throughput Th* is the ratio of the *average throughput* Th_{avg} to network size *N*. Formally, *Th* can be expressed by

$$Th = \frac{Th_{avg}}{N} \tag{14}$$

  Normalized throughput is a good metric for assessing the MIN's cost effectiveness.

- *Relative normalized throughput* of high priority packets *RTh(h)* is the *normalized throughput Th(h)* of high priority packets divided by the *offered load* λ_h of such packets.

$$RTh(h) = \frac{Th(h)}{\lambda_h} \tag{15}$$

- *Relative normalized throughput* of low priority packets *RTh(l)* is the *normalized throughput Th(l)* of low priority packets divided by the *offered load* λ_l of low priority packets.

$$RTh(l) = \frac{Th(l)}{\lambda_l} \tag{16}$$

- *Average packet delay* D_{avg} is the average time a packet spends to pass through the network. Formally, D_{avg} can expressed by

$$D_{avg} = \lim_{u \to \infty} \frac{\sum_{i=1}^{n(u)} t_d(i)}{n(u)} \tag{17}$$

where $n(u)$ denotes the total number of packets accepted within u time intervals and $t_d(i)$ represents the total delay for the ith packet.

- We consider $t_d(i) = t_w(i) + t_{tr}(i)$ where $t_w(i)$ denotes the total queuing delay for i^{th} packet waiting at each stage for the availability of an empty buffer at the next stage queue of the network or for its turn to be transmitted within an SE (the latter applies only to low priority packets which yield to high-priority ones). The second term $t_{tr}(i)$ denotes the total transmission delay for i^{th} packet at each stage of the network; this is equal to $n*nc$, where n is the number of stages and nc is the network cycle.

- *Normalized packet delay D* is the ratio of the D_{avg} to the minimum packet delay which is simply the transmission delay $n*nc$. Formally, D can be defined as

$$D = \frac{D_{avg}}{n * nc} \tag{18}$$

- *Universal performance* (*U)* is defined by a relation involving two above normalized factors, *D* and *Th*: A MIN's performance is considered optimal when *D* is minimized and *Th* is maximized, thus the formula for computing the universal factor arranges so that the overall performance metric for a MIN follows this rule. Formally, *U* can be expressed by

$$U = \sqrt{D^2 + \frac{1}{Th^2}} \tag{19}$$

It is obvious that, when the *packet delay factor* becomes smaller or/and *throughput factor* becomes larger the *universal performance factor* (*U*) becomes smaller. Consequently, as the *universal performance factor* (*U*) becomes smaller, the performance of a MIN is considered to improve. Because the above factors (parameters) have different measurement units and scaling, we normalize them to obtain a reference value domain. Normalization is performed by dividing the value of each factor by the (algebraic) minimum or maximum value that this factor may attain. Thus, equation (19) can be replaced by:

$$U = \sqrt{\left(\frac{D - D^{\min}}{D^{\min}}\right)^2 + \left(\frac{Th^{\max} - Th}{Th}\right)^2} \tag{20}$$

where D^{min} is the minimum value of *normalized packet delay (D)* and Th^{max} is the maximum value of *normalized throughput*. Consistently to equation (19), where the *universal performance factor U*, as computed by equation 20 is close to zero, the MIN performance is considered optimal whereas, when the value of *U* increases, the MIN performance deteriorates. Finally, taking into account that the values of both *delay* and *throughput* appearing in equation (20) are normalized, $D^{\min} = Th^{\max} = 1$, thus the equation can be simplified to:

$$U = \sqrt{(D-1)^2 + \left(\frac{1 - Th}{Th}\right)^2} \tag{21}$$

Finally, we list the major parameters affecting the performance of a MIN.

- *Buffer size* (*b*) is the maximum number of packets that an input buffer of a SE can hold. In our paper we consider a finite-buffered (*b*=1, 2, 3, 4) MIN.
- *Offered load (λ)* is the steady-state fixed probability of arriving packets at each queue on inputs. In our simulation the λ is assumed to be λ = 0.1, 0.2, … 0.9, 1.
- *Ratio of high priority offered load* (r_h), where $r_h = \lambda_h/\lambda$. In our study r_h is assumed to be r_h =0.20 or 0.30.
- *Network size n*, where $n=\log_2 N$, is the number of stages of an (*N* X *N*) MIN. In our simulation *n* is assumed to be *n*=6, 8, 10.

5 Simulation results and discussion

The performance of MINs is usually determined by modelling, using simulation [35] or mathematical methods [36]. In this paper we evaluated the network performance using simulation experiments due to the complexity of the model. For this purpose we developed a special simulator in C++, capable of handling 2-class priority MINs. The simulator has several parameters such as the *buffer-length*, the *number* of input and output ports, the *number* of stages, the *offered load*, and the *ratio* of high priority packets. The simulation was performed at packet level, assuming fixed-length packets transmitted in equal-length time slots, where the slot was the time required to forward a packet from one stage to the next. Each SE was modelled by four non-shared buffer queues, the first two for high priority packets, and the other two for low priority ones. Buffer operation was based on the FCFS principle. The contention between two packets was solved randomly, but when a 2-class priority mechanism was used, high priority packets had precedence over the low priority ones, and contentions were resolved by favouring the packet designated as "high priority" and transmitted from the queue in which the high priority packets were stored in.

The parameters for the packet traffic model were varied across simulation experiments to generate different offered *loads* and traffic patterns. Metrics such as packet *throughput* and packet *delays* were collected at the output ports. We performed extensive simulations to validate our results. All statistics obtained from simulation running for 10^5 clock cycles. The number of simulation runs was adjusted to ensure a steady-state operating condition for the MIN. There was a stabilization process in order the network be allowed to reach a steady state by discarding the first 10^3 network cycles, before collecting the statistics.

5.1 Simulation results for Single priority MINs and Simulator Baselining

Nowadays, in literature there is much work related to a single priority scheme of MINs. The proposed 2-class priority scheme operates also and as a single priority one in the marginal cases, when λ_h=0 or λ_l=0. For these marginal cases, the results we obtained from our simulations coincide with those reported in the literature for single-priority MINS, validating thus the accuracy of our simulator.

Fig. 4 shows the *normalized throughput* of a single priority, single-buffered, 6-stage MIN vs. the *input load* for three classical models [21] [23] [33] and our simulation for the aforementioned marginal cases. All models give identical results only at low *offered loads*, but diverge as the offered load increases. Existing literature [33] has concluded that Theimer's model is the most accurate one: Jenq's model, in particular, looses accuracy because it does not adequately handle the fact that many packets are blocked mainly at the first stages of the MIN at higher traffic rates. The accuracy of Mun's model was also improved considerably by introducing a "blocked" state. Finally, the accuracy of Theimer's model was further (yet marginally) improved in our simulator by considering the dependencies between the upper and low buffers of each SE. Our simulation was tested by comparing the results of the Theimer's model with those obtained by our simulation experiments which were found to be in a close agreement (differences were less than 1%). All models were implemented and ran in the custom-developed C++ simulator described in the beginning of the section 5. We tested various buffer sizes (1, 2, 4) and load ranges varying from λ=0.1 to λ=1, and in all cases the comparison between our model and Theimer's model yielded differences less than 1%.

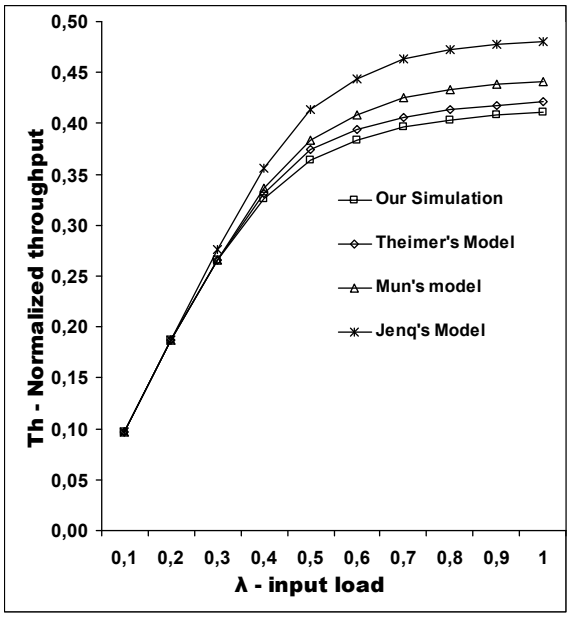

Fig. 4 *Normalized throughput* of a single priority, single-buffered, 6-stage MIN according to different simulation models

5.2 2-class priority MINs vs. single priority ones

In this paper we address the performance evaluation of the 2-class priority scheme for MINs, aiming to get insight on the effects of each factor on the overall performance of this MIN class. In this section, we present our findings and compare different configurations of 2-class priority MINS; we also compare the performance metrics of 2-class priority MINs against the single-priority MIN class.

Fig. 5 illustrates the gains on *total normalized throughput* of a MIN using a 2-class priority scheme versus a single priority one. In the diagram, curve 2P[10]B[*b*]H[20] depicts the *total normalized throughput* of a 2-class priority, 10-stage MIN, under various *buffer-length* setups (*b*=1, 2, 4), when the *ratio* of high priority packets is 20%. Similarly, curve 1P[10]B[*b*] shows the corresponding *normalized throughput of* a single priority, 10-stage MIN, under the same *buffer-length* setups (*b*=1, 2, 4). In this figure, all curves represent the performance factor of *normalized throughput* at different offered loads (λ=0.1, 0.2… 1).

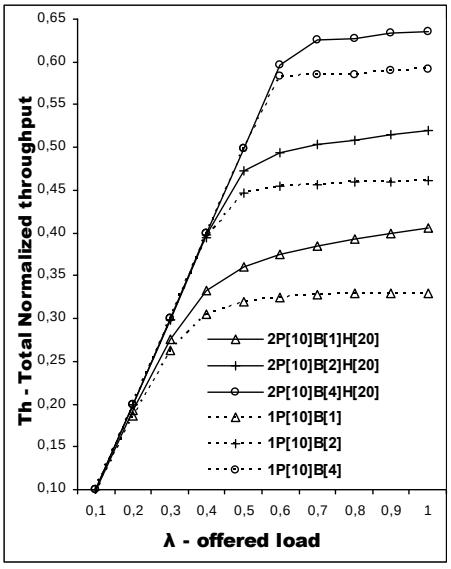

Fig. 5 *Total normalized throughput* of single- and dual-priority 10-stage MINs

We can notice here that the gains on *total normalized throughput* of a 2-class priority scheme for a 10-stage MIN versus a single priority one are 23%, 12.6%, and 7.4%, when the *buffer-lengths* are 1, 2, and 4 respectively,

under a high-priority appearance of 20%, and full load traffic conditions. The throughput gains can be mainly attributed to the exploitation of the extra buffer spaces available in the SEs of the 2-class priority MINs: recall that SEs in single-buffered MIN supporting one priority class have a single buffer available per incoming link; in single-buffer MINs supporting two priorities, however, SEs have one buffer for high-priority packets and one buffer for low-priority packets per input link. The normalized throughput of single-buffer MINs supporting two priorities appears though inferior to that of double-buffered single-priority MINs in fig. 5, because the extra buffer available in dual-priority MINs is exploited only for high-priority packets (20% of the total traffic), and thus remains unexploited when no high-priority packets are available. Contrary to that, double-buffered single-priority MINs can exploit the extra buffer space for any packet, with no restriction whatsoever. In figure 5 we can finally notice that the input load at which the dual-priority MIN's performance starts to have an edge over its single-priority counterpart is smaller for single-buffered MINs ($\lambda \approx 0.3$) and smaller for double-buffered ($\lambda \approx 0.5$) and quad-buffered MINs ($\lambda \approx 0.6$). These loads correspond to the points where the probability that single-priority MIN buffers are full (leading thus to packet blockings) exceeds a certain threshold, having therefore observable effects.

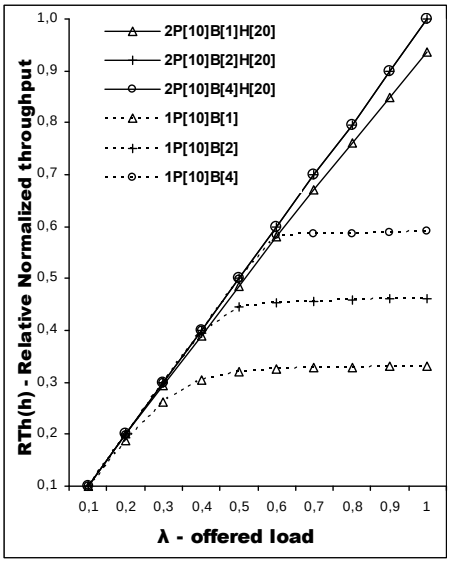

Fig. 6 *Normalized throughput* of high priority packets on a 2-class, finite-buffered, 10-stage MIN

Figure 6, depicts the metric of *relative normalized throughput* for high priority packets in a MIN using the 2-class priority scheme and the (overall) *relative normalized throughput* for single-priority MINs. All measurements apply to a 10-stage MIN, and when packets of two priorities are considered, they account to 20% of the overall traffic; measurements have been collected for buffer lengths b = 1, 2 and 4.

It is worth noting that the *relative normalized throughput* of high priority packets is improved dramatically for all configuration setups, approaching the optimal value (Th_{max}=1), especially when b>=2, under full load traffic conditions. Practically, for b>=2 and under the examined conditions, blockings events for high-priority packets were very rare: this is due to the fact that the network has amble power to optimally service high priority packets, which constitute the 20% of the overall traffic. Buffer space usage analysis in our simulation considering a MIN configured with buffer size b=2 has produced the following results:

- for 74.9% of the network cycles both high-priority buffers were empty; when both high-priority buffers are empty, no blocking can occur, since even if two new high-priority packets arrive (one per incoming link), there is still enough space to store both of them.
- for 23.7% of the network cycles only one high-priority buffer space was occupied and the other was empty; for a blocking to occur at this state, two new high-priority packets must arrive (one per incoming link) *and* the already existing packet must be blocked (because no buffer space is available at the next stage). Under a ratio of high priority offered load equal to 20%, this situation is highly improbable; actually, in the conducted simulations no such blocking was recorded.
- for 1.4% of the network cycles both high-priority buffer spaces were full; for a blocking to occur at this state, either (i) two new high-priority packets must arrive (and thus at least one will be blocked since at most one existing packet will be transmitted to the next stage) or (ii) one new high-priority packets must arrive *and* the transmission of an existing packet must be blocked. In the conducted simulations, type (i) blockings accounted for 0,6% of the respective network cycles (thus an overall 1.4% * 0.6% = 0.0084% of the overall cycles) and type (ii) blockings accounted for another 0.45% of

the respective network cycles (thus an overall 0.0063% of the overall network cycles), therefore both blocking types are limited to 0.0147% of the overall network cycles, which is a very low ratio.

Figure 7 illustrates the *relative normalized throughput* for low priority packets in a MIN using the 2-class priority scheme and the (overall) *relative normalized throughput* for single-priority MINs. Considering the performance curves for MIN pairs (dual-priority and single-priority) with equal buffer sizes (b = 1, 2, 4), we can identify three segments:

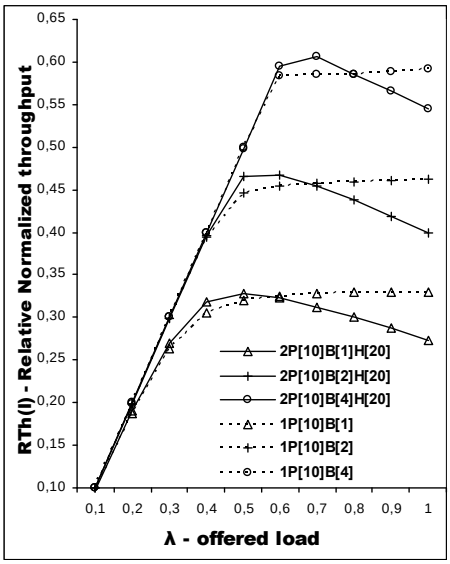

Fig. 7 *Normalized throughput* of low priority packets on a 2-class, finite-buffered, 10-stage MIN

- An initial segment where the performance of single-priority MINs is identical to that of its dual-priority counterpart. This segment corresponds to the load range that the available buffer space in the single-priority MIN is adequate, and blockings are mostly due to packets in the same SE contending for the same output link, rather than due to buffer unavailability at the next MIN stage.
- A middle segment, where the *normalized throughput* of the low-priority packets in the two-priority MIN is superior to the (overall) *normalized throughput* in a single-priority MIN. The beginning of this segment corresponds to the load point where blockings due to buffer unavailability begin to play a part in the MIN performance. In this segment, the gains obtained from the exploitation of the extra buffer space in the dual

priority MINs is higher than the penalization incurred for low-priority packets, due to the fact that they yield to high-priority ones.

- An ending segment, where the *normalized throughput* of the low-priority packets in the dual priority MIN is inferior to the (overall) *normalized throughput* in a single-priority MIN. This corresponds to the load range where the yielding of low-priority packets incurs higher penalty than the gains obtained due to the availability of the extra buffer space. Especially at loads λ close to 1, buffer space for low-priority packets is already saturated and low-priority packets are further delayed because high-priority packets are preferred for transmission, when present.

In all cases, the maximum deterioration recorded is 15.6% for *b*=1, 13% for *b*=2 and 8.48% for *b*=4. This deterioration can be considered as tolerable, especially considering the gains achieved for high-priority packets.

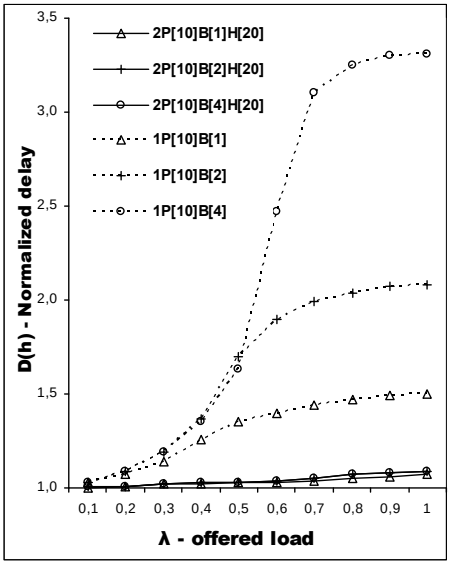

Fig. 8 *Normalized delay* of high priority packets on a 2-class, finite-buffered, 10-stage MIN

Fig. 8 represents the corresponding decrements on *normalized delays* for high priority packets of 2-class priority scheme vs. single priority one for a 10-stage MIN, under a *rate* appearance of 20% for high priority offered loads. It noteworthy, that the improvement of high priority packet delays is considerable for all above

buffer-length configurations of MIN. It follows that *normalized delay* is reduced dramatically to $D_{(h)}$=1.07 … 1.09 approaching the optimal value D_{min}=1. It also follows that the minimization of *normalized delays* for high-priority packets in a 2-class priority scheme is stronger at larger *buffer-length* configurations, where the *packet delays* have greater values in the corresponding single priority MINs.

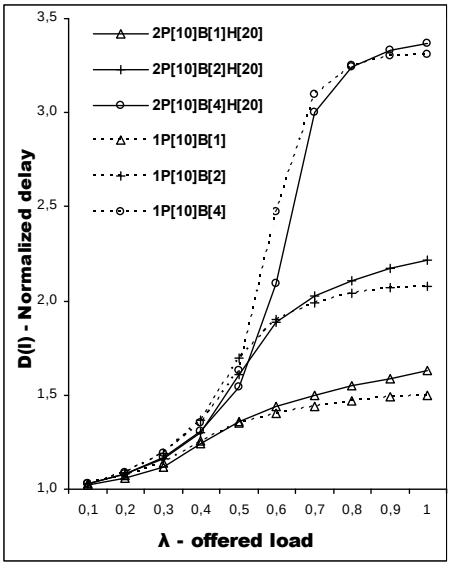

Fig. 9 *Normalized delay* of low priority packets on a 2-class, finite-buffered, 10-stage MIN

Figure 9 illustrates the *normalized delay* for low priority packets in a MIN using the 2-class priority scheme and the (overall) *relative normalized throughput* for single-priority MINs. Similarly to the case of *normalized throughput* for low-priority packets, when examining the performance curves for MIN pairs (dual-priority and single-priority) with equal buffer sizes (b = 1, 2, 4), we can identify three segments:

- An initial segment where the performance of single-priority MINs is identical to that of its dual-priority counterpart. This segment corresponds to the load range that and blockings (which are the cause of the delays) are mostly due to packets in the same SE contending for the same output link, thus the introduction of the two-priority scheme and the extra buffer space in SEs does not alter the observed packet delay in the MIN.

- A middle segment, where the normalized delay of the low-priority packets in the two-priority MIN is superior to the (overall) *normalized throughput* in a single-priority MIN (i.e. has smaller value). This gain stems from the fact that high-priority packets are stored in separate queues in SEs, decreasing thus the number of blockings of low priority packets due to unavailability of suitable buffer space in the destination SE.
- An ending segment, where the *normalized throughput* of the low-priority packets in the two-priority MIN is inferior to the (overall) *normalized throughput* in a single-priority MIN (i.e. it has a larger value). This corresponds to the load range where the yielding of low-priority packets incurs higher penalty than the gains obtained due to the availability of the extra buffer space. Especially at loads λ close to 1, buffer space for low-priority packets is already saturated and low-priority packets are further delayed because high-priority packets are preferred for transmission.

Figures 10 and 11 depict the *relative normalized throughput* for high and low priority packets respectively, in a *k*-stage MIN, where *k*=6,8, and 10, using a 2-class priority scheme, under a packet appearance of 30% for high priority offered load, and full traffic conditions versus the *buffer-length* of MIN. A high-priority packet ratio of 30% was used in these diagrams, to make the effects of the introduction of priority handling more discernible, especially for low priority packets (results for high-priority packets for the 20% ratio case are similar, showing only a slight improvement for b=1). We noticed again that the *relative normalized throughput* of high priority packets is improved dramatically for all *network size* setups, approaching the optimal value (Th_{max}=1), especially when *b*>=2. On the other hand, the loss of *normalized throughput* for corresponding low priority packets ranged from 9% to 24.6%, which is tolerable for all *network size* and *buffer-length* configurations.

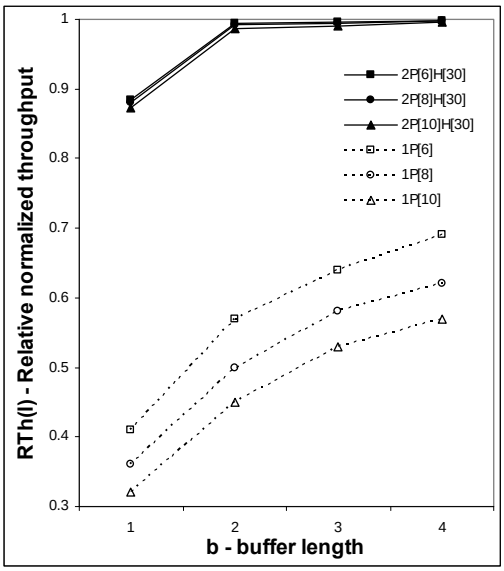

Fig. 10 *Normalized throughput* of high priority packets on a 2-class, finite-buffered, *k*-stage MIN

We can also notice that the *relative normalized throughput* appears to drop as the number of MIN stages increases (for low-priority packets and for single-priority MINs): this happens because although the overall number of packets traversing the network in the unit of time increases along with the number of stages, this increment is less than the theoretical growth of the MIN routing capacity, which the definition of the *relative normalized throughput* takes into account (recall that the normalized throughput metric divides the number of packets traversing the network in the unit of time by the network size, to express the extent to which the MIN's routing capacity is exploited). An equivalent reading of this phenomenon is that fewer packets *per input source* reach their destination per unit of time, when the MIN size increases. This performance degradation is due to the fact that each extra MIN stage introduces an additional point that blockings may occur, mainly due to contentions for the same output link. This is especially true under the full load condition considered in Fig. 11, while for lighter MIN loads, the drop is less observable. MIN designers should take into account this fact when they need to upsize their network installations, and take additional actions if they want to maintain the throughput per input

source; two prominent approaches are the super-linear increase of the network size (leaving some inputs unconnected) and the addition of extra buffer space in the SEs.

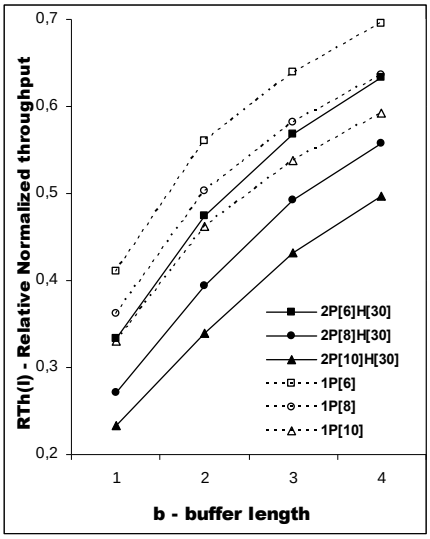

Fig. 11 *Normalized throughput* of low priority packets on a 2-class, finite-buffered, *k*-stage MIN

Figures 12 and 13 illustrate the relation of the combined *performance indicator U* of a 2-class, 10-stage MIN to the *offered load λ,* for high and low priority packets respectively, under different *buffer size* configurations (*b*=1, 2, and 4), when the *ratio* of high priority offered load is 20%. Recall from *section* 3, the combined *performance indicator U* depicts the overall performance of a MIN, considering the weights of each individual performance factor (*throughput* and *packet delay*) are of equal importance. In figure 12 we notice that the value of the *universal performance factor* decreases (thus MIN performance is improved) when the buffer size increases, except for the case of single-priority MINs with *b*=4 and operating under medium and high loads (λ >= 0.6), in which case the *universal performance factor* deteriorates. This holds because the delay in these cases increases rapidly, while gains in the throughput are very small.

In figure 13 we can observe the behaviour of *the universal performance factor* for low priority packets in dual-priority MINs, and the (overall) *universal performance factor* for single-priority MINs when considering different offered loads. Consistently with the respective findings for *normalized throughput* and *delay*, three

segments are identified when examining the performance curves for MIN pairs (dual-priority and single-priority) with equal buffer sizes (b = 1, 2, 4): an initial segment with identical performance among pairs, a middle segment where the dual-priority MIN outperforms the single-priority one and a final segment where the dual-priority MIN lags behind the single-priority one. This is to be expected since the *universal performance factor* combines the individual metrics of *normalized throughput* and *delay*, and since these metrics exhibit a common behaviour, this behaviour is also exhibited in the combined metric.

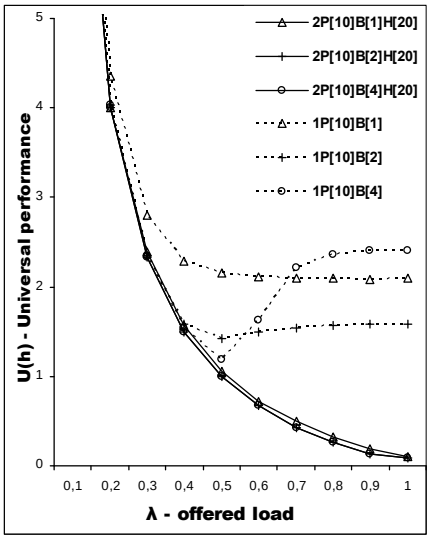

Fig. 12 *Universal Performance* factor of high priority packets on a 2-class, finite-buffered, 10-stage MIN

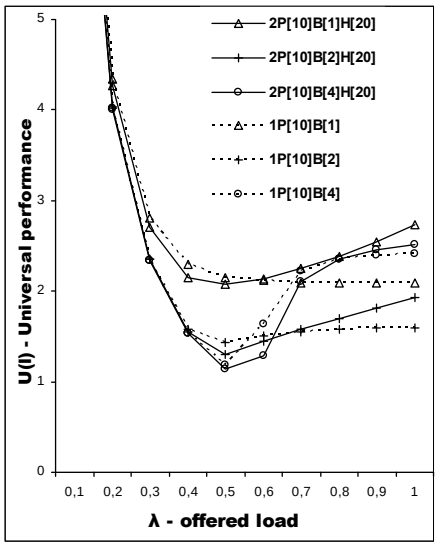

Fig. 13 *Universal Performance* factor of low priority packets on a 2-class, finite-buffered, 10-stage MIN

Regarding the effect of the MIN size on network performance, figures 14-17 illustrate the performance metrics (*relative normalized throughput* and *normalized delay*) for MIN sizes equal to 6, 8, and 10. In this experiment we have fixed the buffer size to 2, since setting the buffer size equal to 1 significantly degrades the MIN performance (cf. fig 11) for low priority packets, while setting the buffer size equal to 4 leads to excessive high delays for low priority packets (cf. fig 9). In particular, figure 14 illustrates the *relative normalized throughput* for high priority packets; as shown in the figure, the size of the MIN has no particular effect on the specific performance metric, with a negligible deterioration (less than 0.7% in all cases) being observed when the MIN size increases from 6 to 10 stages. Similarly, figure 15 illustrates the *normalized delay* for high priority packets; again the deterioration observed when the size of the MIN increases from 6 to 10 stages is very small (less than 1.3% in all cases). This indicates that the network has amble switching power to optimally serve high-priority packets; when the MIN size increases, the number of points within the network where blockings can occur also increases, and this justifies the slight deterioration in both performance metrics. As compared to the results presented in [37], which considers the same MIN sizes (6, 8, 10) but with a single buffer space, it appears that the addition of one extra

buffer space only slightly improves the throughput (approximately 3% at full load), with a corresponding deterioration in the delay (approximately 4% at full load).

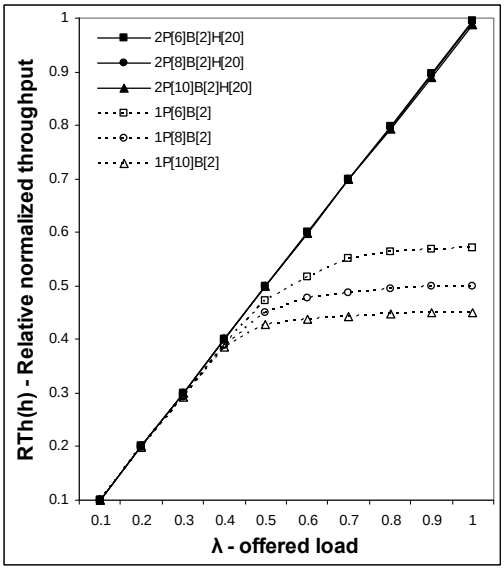

Fig. 14 *Relative normalized throughput* of high-priority packets on 2-class, double-buffered MIN with varying numbers of stages

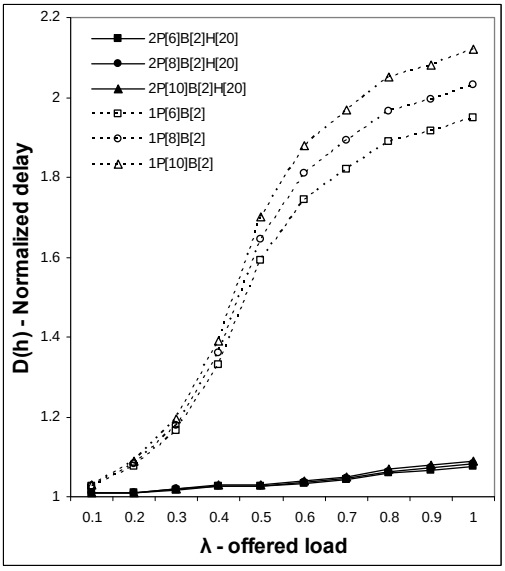

Fig. 15 *Normalized delay* of high-priority packets on 2-class, double-buffered MIN with varying numbers of stages

Figures 16 and 17 depict the *relative normalized throughput* and the *normalized delay*, respectively, for low priority packets. Similarly to the case for high-priority packets, we can observe that both performance factors drop when the MIN size increases. In particular, the *relative normalized throughput* of low priority packets drops by 10.4% at full load, when the MIN size increases from 6 to 10, whereas the corresponding deterioration (increase) for the *normalized delay* of low priority packets is 4.5%. These performance indicator deteriorations are owing to the fact that when the MIN size increases, the number of points within the network where blockings can occur also increases (as also stated above); however the deterioration for low priority packets is significantly higher than for high priority ones, since the serving of the latter packet class (high priority packets) has precedence over the serving of low-priority packets, hence the overall MIN performance degradation mainly affects low priority packets and not high-priority ones.

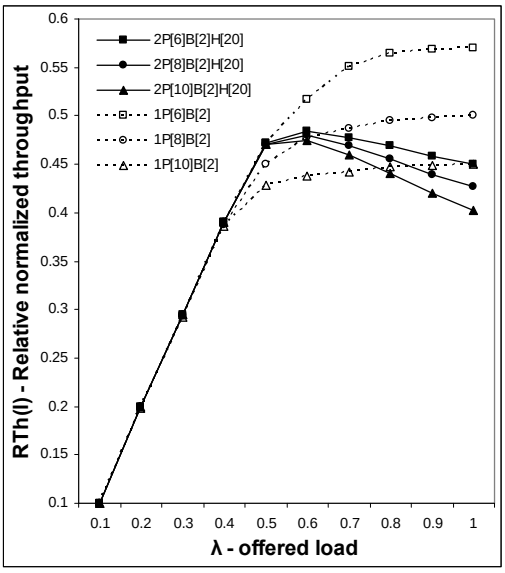

Fig. 16 *Relative normalized throughput* of low-priority packets on 2-class, double-buffered MIN with varying numbers of stages

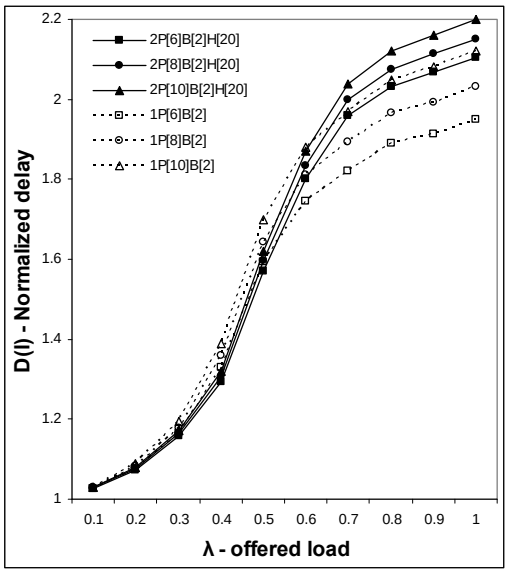

Fig. 17 *Normalized delay* of low-priority packets on 2-class, double-buffered MIN with varying numbers of stages

Regarding the effect of the high/low priority traffic ratio, figures 18-21 illustrate the performance metrics (*relative normalized throughput* and *normalized delay*) for a 2-class, double buffered 10-stage MIN for varying high/low priority ratios. Similarly to the experiments regarding the MIN size, we have fixed the buffer size to b=2, since this value balances the *relative normalized throughput* and *normalized delay* performance indicators for low priority packets. For conciseness purposes, only results regarding 10-stage MINs are presented here, however the results for 6- and 8-stage MINs are analogous.

More specifically, figure 18 illustrates the *relative normalized throughput* of high-priority packets; we can observe that high priority packets are served close to optimally in all cases, albeit a deterioration of 3.34% is observed when the ratio of high-priority packets increases from 10% to 30%. Figure 19 depicts the *normalized delay* for high priority packets; *normalized delay* remains at acceptable levels when the ratio of high-priority packets rises to 30% (1.15 against an optimal value of 1), however the deterioration against the case of the 10% high priority ratio is considerable (10.6%). The increase in the delay is mainly owing to contentions for transmission through the same output link.

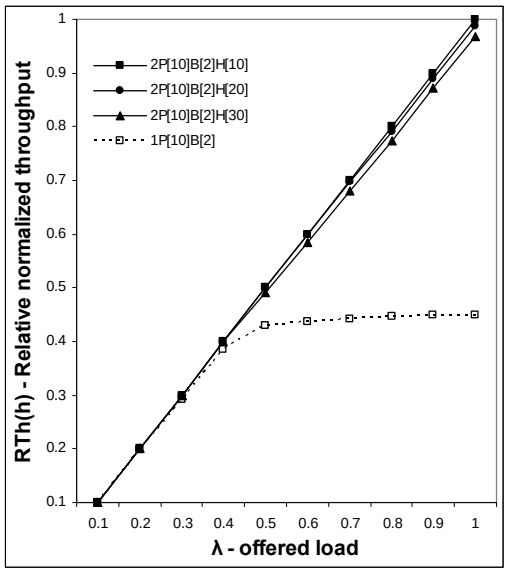

Fig. 18 *Relative normalized throughput* of high-priority packets on 2-class, double-buffered, 10-stage MIN for varying high/low priority ratios

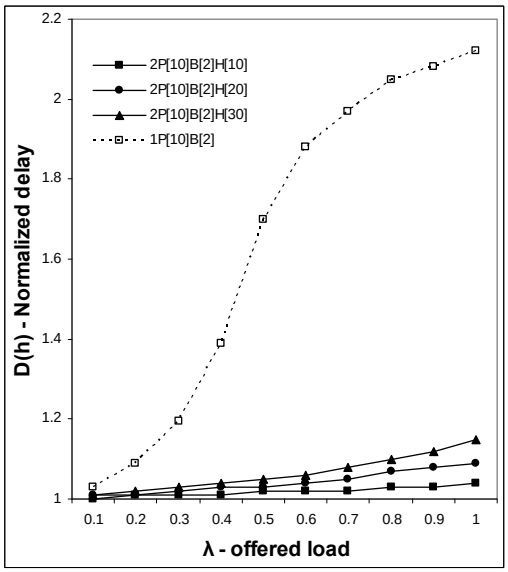

Fig. 19 *Normalized delay* of high-priority packets on 2-class, double-buffered, 10-stage MIN for varying high/low priority ratios

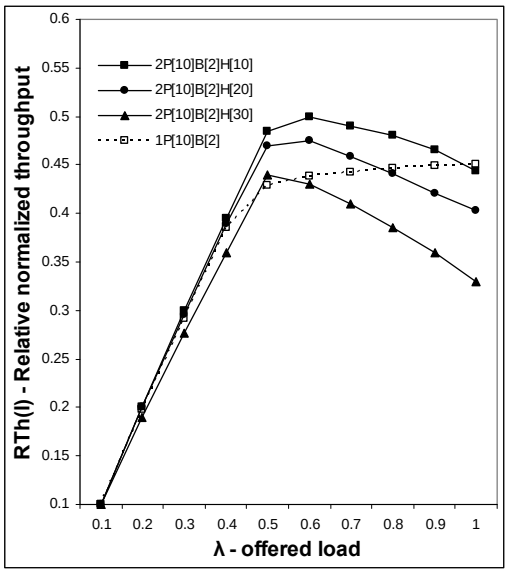

Fig. 20 *Relative normalized throughput* of low-priority packets on 2-class, double-buffered, 10-stage MIN for varying high/low priority ratios

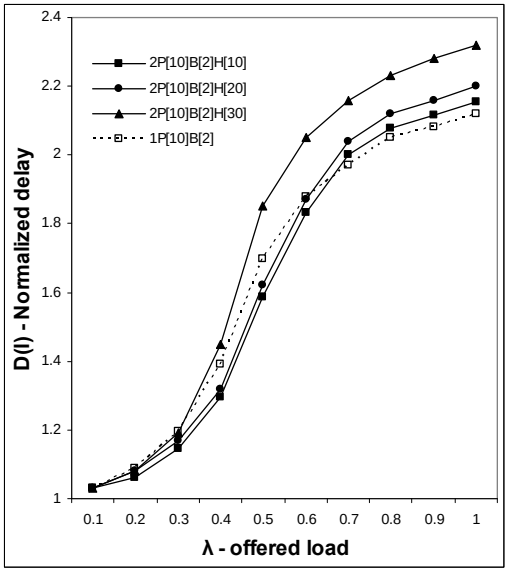

Fig. 21 *Normalized delay* of low-priority packets on 2-class, double-buffered, 10-stage MIN for varying high/low priority ratios

Figures 20 and 21 illustrate the *relative normalized throughput* and the *normalized delay* for low priority packets. Expectedly, when the high/low priority ratio increases, the performance indicators for low priority

packets decline, since the probability for them to contend for the same output link with a high-priority packet -and yield to it- is higher. Similarly to the case of figure 7, when comparing the dual-priority curves to the single-priority one, we can identify an initial segment where the *relative normalized throughput* is identical (the network has enough switching power to serve all packets); then a middle segment where the two-priority scheme exhibits higher *relative normalized throughput* than the single priority one (the benefits from having an extra buffer space for low-priority packets exceed the losses from yielding to high-priority ones); and an ending segment, where the losses from yielding to high-priority packets surpass the benefits from the extra buffer space, hence at that segment the *relative normalized throughput* observed for the dual-priority scheme is inferior to the single-priority one. Finally, in figure 21 we can observe that higher high/low priority ratios lead to increased delays for low-priority packets; this is to be expected since when the high/low priority ratio increases, the probability for a low-priority packet to yield to a high-priority one rises. This is particularly observable at high loads, where the delay rises by 7.6% as the high/low priority ratio increases from 10% to 30%.

6 Conclusions

In this paper we have addressed the performance evaluation of 2-class priority MINs. We have modelled an analytical system of equations, employing a scheme that takes into account both the previous and the last state of the switching elements, providing thus better accuracy than schemes considering only the last state. We have also evaluated the performance of 2-class priority MINs under varying *offered loads* and *buffer sizes*, considering the high-priority and low-priority packet classes, as well as cumulative performance for the MIN, and compared these metrics against the corresponding performance figures of single-priority MINs. In this study, we have taken into account the two most important network performance metrics namely *throughput* and *packet delay*. The diagrams and discussions given may be used by network designer to tune parameters for their installations so as to obtain optimal performance for the communication requirements of their environments.

In our future work we intend to have additionally a closed form solution for the above system equations providing thus an analytical mode for the MIN. The introduction of an adaptive scheme, altering buffer allocation

to different priority classes according to current traffic load and high/low priority *ratios* will be investigated as well.

References

[1] Allied Telesis. AT-80xx Series Class of Service COS. Allied Telesis, 2010. http://alliedtelesis.custhelp.com/cgi-bin/alliedtelesis.cfg/php/enduser/std_adp.php?p_faqid=2399

[2] M. Atiquzzaman, C.K. Chen, Realistic modeling of blocked packets for accurate performance evaluation of multistage ATM switches, IEE Proceedings: Communications, 146(4) (1999), pp. 213-221.

[3] Cisco Systems, http://www.cisco.com/en/US/prod/collateral/routers/ps5763/prod_brochure0900aecd800f8118.pdf , 2010.

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

[5] Cisco. Catalyst 6500 Release 15.0SY Software Configuration Guide. CISCO systems, 2011.

[6] Cisco. Catalyst 2960 Switch Software Configuration Guide. CISCO, 2010.

[7] Cisco. Catalyst 4500 Release IOS-XE 3.1.0 SG. Software Configuration Guide. CISCO systems, 2011.

[8] J.S.C. Chen and R. Guerin. Performance study of an input queueing packet switch with two priority classes. IEEE Trans. Commun. 39(1) (1991) pp. 117–126.

[9] Chang-hoon Choi, Sung-chun Kim. Hierarchical multistage interconnection network for shared-memory multiprocessor system. Proceedings of the 1997 ACM Symposium on Applied Computing (1997) pp. 468-472.

[10] D-Link. DES-3250TG 10/100Mbps managed switch. Dlink 2006.

[11] D-Link. User manual for the DES-1024D 24-port 10/100Mbps Fast Ethernet Switch. D-Link, 2005, ftp://ftp.dlink.fr/des/DES-1024D/Manuel/DES-1024D_C1_Manual_v3.00.pdf

[12] D-Link. D-Link™ DGS-1008D Gigabit Ethernet Switch manual. DLink 2007.

[13] Elizabeth Suet Hing Tse. Switch fabric architecture analysis for a scalable bi-directionally reconfigurable IP router. Journal of Systems Architecture: the EUROMICRO Journal, 50(1), (2004) pp. 35-60.

[14] S. Feit. Local area high speed networks.MTP, Indianapolis, 2000.

[15] J. Garofalakis, and E. Stergiou, An analytical performance model for multistage interconnection networks with blocking, Proceedings of Communication Networks and Services Research Conference CNSR 2008, IEEE Press (2008) pp. 247-261.

[16] J. Garofalakis, E. Stergiou, An approximate analytical performance model for multistage interconnection networks with backpressure blocking mechanism, Journal of Communications (JCM), Academy 5(3) (2010) pp. 247-261.

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

[18] Hewlett-Packard, HP 2615-8-PoE Switch QuickSpecs. Hewlett-Packard, 2011.

[19] IEEE 802.1Q Working Group. IEEE Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks. IEEE 2005, http://standards.ieee.org/about/get/802/802.1.html

[20] Intel Corporation. Intel Express 530T standalone switch., Intel 2010.

[21] Y.C.Jenq. Performance analysis of a packet switch based on single-buffered banyan network IEEE Journal Selected Areas of Communications, 1(6), (1983) pp. 1014-1021.

[22] T. Lin, L. Kleinrock. 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 (1991) pp. 68 - 78.

[23] H. Mun and H.Y. Youn. Performance analysis of finite buffered multistage interconnection networks IEEE Transactions on Computers, 43(2), (1994) pp. 153-161.

[24] Net Insight AB. Ethernet Switching for the Gigabit Ethernet Access Module. Net Insight AB, 2011. http://www.netinsight.net/Global/Documents/Products/PDS_ethernet_switching_feature.pdf?epslanguage=sv

[25] Netgear. Quality of Service (QoS) on Netgear switches. Netgear, 2009.

[26] Network Working Group. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. IETF, 1998. http://tools.ietf.org/html/rfc2474

[27] S. L. Ng and B. Dewar, Load sharing replicated buffered banyan networks with priority traffic Connecting the System: Australian Telecommunication Networks and Application Conference, Monash University, Clayton, Victoria (1995) pp. 77-82.

[28] J.H. Patel. Processor-memory interconnections for mutliprocessors. Proceedings of 6th Annual Symposium on Computer Architecture New York (1979) pp. 168-177.

[29] M. Saleh, M. Atiquzzaman, Analysis of Shared Buffer Switches under Nonuniform Traffic Pattern and Global Flow Control, Computer Networks, 34(2) (2000) pp. 297–315.

[30] M. Saleh, M. Atiquzzaman, An Exact Model For Analysis of Shared Buffer ATM Switches With Arbitrary Traffic Distribution, IEE Proceedings - Communications, 148(2) (2001) pp. 63-69.

[31] E. Stergiou, J. Garofalakis, Performance Estimation of Banyan Semi Layer Networks with Drop Resolution mechanism, Journal of Network and Computer Applications, Elsevier, 35(1) (2012) pp. 287-294.

[32] Stevens W. R., TCP/IP Illustrated: Volume 1. The protocols, 10th Edition, Addison-Wesley Pub Company (1997).

[33] T.H. Theimer, E. P. Rathgeb and M.N. Huber. Performance Analysis of Buffered Banyan Networks. IEEE Transactions on Communications, 39(2) (1991) pp. 269-277.

[34] Josep Torrellas, Zheng Zhang. The Performance of the Cedar Multistage Switching Network. IEEE Transactions on Parallel and Distributed Systems, 8(4) (1997) pp. 321-336.

[35] D. Tutsch, M.Brenner. MIN Simulate. A Multistage Interconnection Network Simulator. 17th European Simulation Multiconference: Foundations for Successful Modelling & Simulation (ESM'03); Nottingham, SCS pp. 211-216, 2003.

[36] D.Tutsch, G.Hommel. Generating Systems of Equations for Performance Evaluation of Buffered Multistage Interconnection Networks. Journal of Parallel and Distributed Computing, 62(2) (2002) pp. 228-240.

[37] D.C. Vasiliadis, G.E. Rizos, C. Vassilakis, E. Glavas. Performance evaluation of two-priority network schema for single-buffered Delta Networks. Proceedings of the 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2007), IEEE press (2007), pp. 1-7.

[38] D.C. Vasiliadis, G.E. Rizos, and C. Vassilakis. Performance Analysis of dual priority single-buffered blocking Multistage Interconnection Networks, Proceedings of the International Conference on Networking and Services (ICNS '07), IEEE Press (2007), pp. 60-65.

[39] D.C. Vasiliadis, G.E. Rizos, and C. Vassilakis. Routing and Performance Evaluation of Dual Priority Delta Networks under Hotspot Environment, Proceedings of the First International Conference on Advances in Future Internet AFIN09, IEEE Press (2009) pp. 24-30.

[40] Westermo. R200 RingSwitch feature page. Westermo, 2011, http://www.westermo.net/Resource.phx/content/products/ethernet/ontime/ringswitch.htx

[41] B.Zhou, M.Atiquzzaman. A Performance Comparison of Four Buffering Schemes for Multistage Interconnection Networks. International Journal of Parallel and Distributed Systems and Networks, 5(1) (2002), pp. 17-25.