Performance Analysis of Dual-Priority Multilayer Multistage Interconnection Networks under Multicast Environment

Dimitris C. Vasiliadis[a,b], George E. Rizos[a,b] and Costas Vassilakis[a]
[a] *Department of Computer Science and Technology, University of Peloponnese, Tripoli, Greece*
[b] *Technological Educational Institute of Epirus, Arta, Greece*
Email: {*dvas, georizos, costas*}*@uop.gr*

***Abstract*—Next-generation network architectures strive to achieve high bandwidth and ultralow latency for the packets traversing the offered end-to-end paths. Multistage Interconnection Networks (MINs) are often employed for implementing NGNs, but while MINs are fairly flexible in handling varieties of traffic loads, they tend to quickly saturate under broadcast and multicast traffic, especially at increasing size networks. As a response to this issue, multilayer MINs have been proposed, however their performance prediction and evaluation has not been studied sufficiently insofar. In this paper, we evaluate and discuss the performance of multilayer MINs under multicast traffic, considering also two levels of packet priorities, since support for multiple QoS levels is an indispensible requirement for NGNs. Different offered loads and buffer size configurations are examined in this context, and performance results are given for the two most important network performance factors, namely packet throughput and delay. We also introduce and calculate a universal performance factor, which includes the importance aspect of each of the above main performance factors. The findings of this study can be used by NGN system designers in order to predict the performance of each configuration and adjust the design of their communication infrastructure to the traffic requirements at hand.**

***Index Terms*—performance analysis, multistage interconnection networks, banyan networks, multicast traffic, multilayer networks**

I. Introduction

Convergence in network technologies services and in terminal equipment is at the basis of change in innovative offers and new business models in the communications sector [1]. At the network level, this convergence must be supported by low-latency, high-throughput, QoS-aware, packet-switched communication infrastructures, and Multistage Interconnection Networks (MINs) have been recognized as an infrastructure capable of delivering the characteristics above. The switching fabric that provides the communications path between line cards is 3-stage, self-routed architecture. MIN technology is a prominent candidate for the implementation of NGNs, due to its ability to route multiple communication tasks concurrently and the appealing cost/performance ratio it achieves. MINs with the Banyan [2] property, in particular, e.g. Delta Networks [3], Omega Networks [4], and Generalized Cube Networks [5] are more widely adopted, since non-Banyan MINs have generally higher cost and complexity. An example of a MIN-based NGN infrastructure is the, CISCO CRS-1 router [6], which has been built as a multistage switching fabric. MIN-based solutions for NGNs have been widely deployed, as reported in [7].

Multicasting and broadcasting are two important functionalities of communication infrastructure in general and NGNs in particular. This has been recognized by international bodies, such as ITU, which has included multicasting in the NGNs' functional requirements [8] and outlined the framework for the multicast service in NGNs [9]. Existing performance analyses regarding MINs, however, have shown that they quickly saturate under broadcast and multicast traffic [10][11]: within a MIN with the Banyan property, multicasting/broadcasting is implemented through packet cloning [12][13][14], and the MIN switching fabric is unable to efficiently handle the increased number of packets.

As a response to this problem, the replication of the whole MIN network or certain stages of it has been suggested, leading to *multi-layer MINs* [15]. The degree of replication L may be constant for all stages or vary across stages; in general, higher replication degrees should be employed towards the later stages of the MIN to provide the increased switching capacity needed there due to the cloned packets. Since layer replication increases the hardware cost, the first MIN layers are either not replicated at all or replicated at a modest degree, to keep the MIN manufacturing cost at modest levels.

In this paper, we extend previous studies in the area of performance evaluation of MINs (e.g. [16], [17]) by including multi-layer MINs under multicast traffic. We applied the partial multicast policy [18], since it offers superior performance compared to the full multicast mechanism - where a packet is copied and transmitted when only both destination buffers are available - as shown in

previous studies, e.g. [19]. Furthermore, we extend the study presented in [19] by considering Switching Elements (SEs) that natively support a dual priority scheme, and also considering double-buffered SEs, instead of only single-buffered ones; double-buffering has been reported to provide better QoS for packets, as compared to single-buffering [18].

The remainder of this paper is organized as follows: in section 2 we briefly analyze a multilayer MIN for supporting multicasting routing traffic. Subsequently, in section 3 we present the configuration and operational parameters considered in this paper, whereas in section 4 we present the performance evaluation metrics that are collected. Section 5 presents the results of our performance analysis, which has been conducted through simulation experiments; in this section we briefly discuss multicasting in single-layer MINs under a two priority scheme, and then we extend our discussion to cover multi-layer MINs. We also consider a special case of MIN multicasting, according to which multicasting occurs only in the last stages of the MIN, a setup that may occur when cases that LAN switches and trunk networks converge to a single packet routing device. Finally, section 6 provides the concluding remarks.

II. MULTILAYER, MULTI-PRIORITY MIN DESCRIPTION

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. All types of blocking Multistage Interconnection Networks (Delta Networks [3], Omega Networks [4] and Generalized Cube Networks [5]) with the Banyan property which is defined in [2] are 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. Consequently, 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.

A typical configuration of an (NXN) Omega Network is depicted in figure 1 and outlined below. At this paper we also extend previous studies by considering multi-layer MINs, where the lateral view of a typical configuration of (8X8) multi-layer MIN is depicted at figure 2. The example network consists of two segments, an initial single-layer one and a subsequent multi-layer one (with 2 layers). Each SE is modeled by an array of p non-shared buffer queue pairs; within each pair, one buffer is dedicated for the upper queuing bank and the other for the lower bank. During a single network cycle, the SE considers all its input links, examining the buffer queues in the arrays in decreasing order of priority. If a queue is not empty, the first packet from it is extracted and transmitted towards the next MIN stage; packets in lower priority queues are forwarded to an SE's output link only if no packet in a higher priority queue is tagged to be forwarded to the same output link.

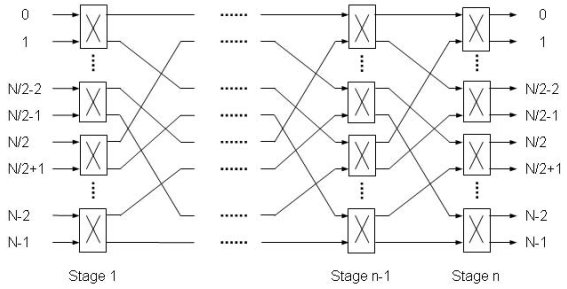

Figure 1. A (NXN) Omega Network

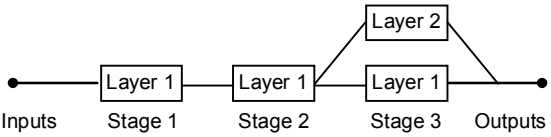

Figure 2. A lateral view of an (8X8) multi-layer MIN

According to figure 2, it is worth noting that packet forwarding from stage 2 to stage 3 is blocking-free, since packets in stage-2 SEs do not contend for the same output link; packets at this stage can also be "cloned" (i.e. forwarded to both subsequent SEs in the context of a multicast routing activity), again without any blocking. This is always possible for cases where the degree of replication of succeeding stage i+1 (which we will denote as l_{i+1}) is equal to 2*l_i. If, for some MIN with n stages there exists some nb ($1 \le nb < n$) such that $\forall$k: $l_{k+1} = 2 * l_k$ ($nb \le k < n$), then the MIN operates in a non-blocking fashion for the last ($n - nb$) stages. Note that according to [15], blocking can occur at the MIN outputs, where SE outputs are multiplexed, if either the multiplexer or the data sink do not have enough capacity; in this paper however we will assume that both multiplexers and data sinks have adequate capacity.

In this paper, we consider dual-priority, finite-buffered, Multistage Interconnection Networks supporting multicast traffic which operate 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. Internal clocking results in synchronously operating switches in a slotted time model [20], and all SEs have deterministic service time.
- 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 $\lambda = \lambda_h + \lambda_l$.

- At each input of the network only one packet can be accepted within a time slot. All packets in input ports contain both the data to be transferred and the routing tag. The priority of each packet is indicated through a priority bit in the packet header. Under the dual-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 (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.
- The packet header consist of two extra equal-length fields the *Routing Address* (RA) and *Multicast Mask* (MM), occupying n bits each, where n is the number of stages in the MIN. Upon reception of a packet, the SE at stage k first examines the k^{th} bit of the MM; if this is set to 1, then the packet makes a multicast instead of a unicast transmission, forwarding the packet to both its output links. If the k^{th} bit of the MM is however set to zero, then the k^{th} bit of the RA is examined, and routing is performed as in the case of unicast MINs. It is obvious that, when all bits of the MM of a packet are set zero, the packet follows a unicast path, reaching one specific network output port. On the other extreme, when all its bits are set to one the packet is broadcasted to all output ports of the network. In all other cases, the packet will be forwarded to a group of output ports, which constitute the *Multicast Group* (MG).
- 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. Since 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.
- Finally, packets are removed from their destinations immediately upon arrival, thus packets cannot be blocked at the last stage.

III. Configuration and Operational Parameters of the Evaluated MINs

In this paper we extend our study on performance evaluation of MINs by comparing the performance of dual-priority architecture versus single-priority one under multicast traffic. All presented MINs are constructed by either single- or multi-layer SEs. According to fig. 1 the proposed MINs consists of two segments, the first one having only single-layer SEs, and the second one which is multi-layer. In the multi-layer segment each stage $i+1$ has twice as many layers as the immediately preceding one, i, thus this segment operates in a non-blocking manner. This stems from the fact that if stage i has n_i SEs of l_i layers each, at a certain MIN network cycle *at most* $2^* n_{i^*} l_i$ packets will be generated by this stage (all SEs at all layers process and clone a packet), and the subsequent one has enough SEs to intercept and process these packets. Consequently, all SEs at multi-layer segment are considered to have only the buffer space needed to store and forward a single packet. On the other hand, the SEs of single-layer segment may employ different *buffer sizes* in order to improve the overall MINs performance. Under these considerations, the operational parameters of the MINs evaluated in this paper are as follows:

Buffer-size b of a queue is the maximum number of packets that an input buffer of a SE can hold. In this study both symmetric single- and double-buffered MINs (b = 1, 2) are considered. We note here that the particular *buffer sizes* have been chosen since they have been reported [21] to provide optimal overall network performance: indeed, [21] documents that for higher *buffer sizes* (b = 4, 8) *packet delay* increases significantly, while SE hardware cost is also elevated. Furthermore, in the case of multilayer MINs the balance between the overall performance and cost is of crucial importance, since the addition of layers leads to a rising cost.

Offered load λ is the steady-state fixed probability of such arriving packets at each queue on inputs. In our simulation λ is assumed to be λ = 0.1, 0.2… 0.9, 1. 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 $\lambda = \lambda_h + \lambda_l$.

Ratio of high priority offered load r_h, is defined by $r_h = \lambda_h/\lambda$. In our study r_h is assumed to be $r_h = 0.10$. Similarly, the ratio of low priority offered load r_l can be expressed by $r_l = \lambda_l/\lambda$. It is obvious that that $r_h + r_l = 1$. Consequently, r_l is assumed to be $r_l = 0.90$.

Number of stages n, is the number of stages of an (N X N) MIN, where $n=\log_2 N$. In our simulation n is assumed to be $n=6$, which is a widely used MIN size.

Multicast ratio m of a SE at stage k, where $k=1\ldots n$ is the probability of a packet having the k^{th} bit of its multicast mask (MM) set 1. Consequently, it effectively expresses the probability that this particular SE will do a multicast by forwarding the packet to both its output links. In this paper m is considered to be fixed at all SEs and is assumed to be m = 0, 0.1, 0.5. It is obvious that, when m = 0 all input traffic is unicast, while if $m=1$ all packets are broadcast (i.e. all packets reach all destinations). For intermediate values of m, the probability that a packet is unicast is equal to $(1-m)^n$, i.e. the joint probability that all bits in MM are equal to 0. The value m=0.1 for multicast ratio is considered, since it for a MIN size n equal to 6 evaluates to $(1\text{-}0.1)^6 = 0.9^6$ =53.14%, giving thus approximately equal probabilities for unicast or multicast transmission within the MIN.

IV. Performance Evaluation Metrics for MINs

The two most important network performance factors, namely *packet throughput* and *delay* are evaluated and analyzed in this section. The *Universal performance factor* introduced in [21], which combines the above two metrics into a single one is also applied. In this study, when calculating the value of this combined factor, we have considered the individual performance factors (*packet throughput* and *delay*) to be of equal importance. This is not necessarily true for all application classes, e.g. for batch data transfers *throughput* is more important, whereas for streaming media the *delay* must be optimized. Moreover, attention has been paid to the definition of *throughput* and *delay* for multi-layer MINs, since both the single-layered and multi-layered segments have to be considered. Finally, in this work *packet loss probability* is considered as a separate metric for multicast traffic.

A. Metrics for Single-layer MINs

In order to evaluate the performance of a multicasting, single-layer (NXN) MIN the following metrics are used. Let Th and D be the *normalized throughput* and *normalized delay* of a MIN.

Relative normalized throughput RTh(h) of high priority packets is the normalized throughput Th(h) of such packets divided by the corresponding ratio of offered load r_h.

$$RTh(h) = \frac{Th(h)}{r_h} \tag{1}$$

Similarly, r***elative normalized throughput RTh(l)*** of low priority packets can be expressed by the ratio of *normalized throughput Th(l)* of such packets to the corresponding *ratio of offered load* r_l .

$$RTh(l) = \frac{Th(l)}{r_l} \tag{2}$$

This extra normalization of both high and low priority traffic leads to a common value domain needed for comparing their absolute performance values with those obtained by the corresponding single priority MINs. Thus, in the diagrams of the next section we will compare the *relative normalized throughput* of dual-priority MINs with the *normalized throughput* of single-priority ones.

Universal performance factor Upf is defined by a relation involving the two major above normalized factors, D and Th: the performance of a MIN 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 follows that rule. Formally, *Upf* can be expressed by

$$Upf = \sqrt{w_d * D^2 + w_{th} * \frac{1}{Th^2}} \tag{3}$$

where w_d and w_{th} denote the corresponding *weights* for each factor participating in the *Upf,* designating thus its importance for the corporate environment. Consequently, the performance of a MIN can be expressed in a single metric that is tailored to the needs that a specific MIN setup will serve. It is obvious that, when the *packet delay* factor becomes smaller or/and *throughput* factor becomes larger the *Upf* becomes smaller, thus smaller *Upf* values indicate better overall MIN performance. 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 (3) can be replaced by:

$$Upf = \sqrt{w_d * \left(\frac{D - D^{\min}}{D^{\min}}\right)^2 + w_{th} * \left(\frac{Th^{\max} - Th}{Th}\right)^2} \tag{4}$$

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 (3), when the *universal performance factor Upf*, as computed by equation (4) is close to 0, the performance a MIN is considered optimal whereas, when the value of *Upf* increases, its performance deteriorates. Moreover, taking into account that the values of both *delay* and *throughput* appearing in equation (4) are normalized, $D^{\min} = Th^{\max} = 1$, thus the equation can be simplified to:

$$Upf = \sqrt{w_d * (D-1)^2 + w_{th} * \left(\frac{1 - Th}{Th}\right)^2} \tag{5}$$

In the remaining of this paper we will consider both factors of equal importance, setting thus $w_d = w_{th} = 1$. Finally, as in the evaluation of *relative normalized throughput* for high and low priority traffic the corresponding *ratio of offered load* takes also place at this metric. Consequently,

$$Upf(p) = \sqrt{w_d * (D(p)-1)^2 + w_{th} * \left(\frac{1-Th(p)}{Th(p)}\right)^2} * r_p \quad (6)$$

where $r_p = \{r_h, r_l\}$ is the corresponding *ratio of offered load* for high and low priority traffic respectively.

Average packet loss probability $Pl_{avg}(p)$ is the average number of p-class packets rejected by all input ports per network cycle. Formally, $Pl_{avg}(p)$ is defined as

$$Pl_{avg}(p) = \lim_{u \to \infty} \frac{\sum_{k=1}^{u} n_r(p,k)}{u} \quad (7)$$

where $n_r(p,k)$ denotes the total number of p-class packets that are rejected at all queues of SEs at the first stage of MIN during the k^{th} time interval.

Normalized packet loss probability Pl(p) is the ratio of the *average packet loss probability* $Pl_{avg}(p)$ to the number of network input ports N. As in equations (1) and (2) the *relative normalized loss probability* $Pl(p)$ can be formally expressed by

$$Pl(p) = \frac{Pl_{avg}(p)}{N * r_p} \quad (8)$$

Note that the *packet loss probability* in the case of unicast traffic is equal to (λ-Th), and this is the reason it does not appear in the Upf formula in [21] (this paper considers only unicast traffic). In this work, we will retain the definition of [21] for Upf, and we will consider *packet loss probability* as a separate metric for multicast traffic.

B. Metrics for multi-layer MINs

Recall from section 3 that multilayer (N X N) MINs considered in this paper consist of two segments, as illustrated in figure 1: the first one is a single-layer segment and the second one is a multi-layer segment operating in a non-blocking fashion. Let l be the number of layers at the last stage (output) of network. The number of multi-layer stages is then $n_{ml} = \log_2 l$ (since layers are doubled in consecutive stages in the multilayer segment), while the number of single-layer stages is $n_{sl} = n - \log_2 l = \log_2 N - \log_2 l$, where $n = \log_2 N$ is the total number of stages in the MIN.

Normalized throughput Th(p) of an l-layer MIN can be consequently expressed as

$$Th(p) = Th(p, n - \log_2 l) * (1+m)^{1+\log_2 l} \quad (9)$$

where $Th(p,n\text{-}\log_2 l)$ is the *normalized throughput* of p-class traffic at last stage of single-layer segment of MIN. The multiplier in equation (9) $[(1+m)^{1+\log_2 l}]$ effectively represents the *cloning factor* of a packet undergoing $1+\log_2 l$ transmissions across stages, with the probability of being duplicated in each transmission is m. Note that equation (9) holds under the assumption that no blockings may occur in the last $1+\log_2 l$ transmissions; the last one of single-layer and all of multi-layer segment.

Normalized delay D(p) of an l-layer MIN can be similarly evaluated basing on the *normalized packet delay* $D(p,n\text{-}\log_2 l)$ of single-layer segment of MIN. Formally, $D(p)$ can be defined as

$$D(p) = \frac{D(p, n-\log_2 l) * (n-\log_2 l) + \log_2 l}{n} \quad (10)$$

The *normalized delay* of entire MIN transmission includes both single- and multi-layer segments. The *average delay* of the single-layer segment can be expressed as $D_{avg}(p,n-\log_2 l) = D(p,n\text{-}\log_2 l) * (n\text{-}\log_2 l) * nc$. Subsequently, the *average delay* $D_{avg}(p)$ of entire l-layer MIN is simply augmented by the transmission delay of non-blocking, multi-layer segment which is $\log_2 l * nc$. Thus, the *normalized delay* just as expressed by equation (10) is computed by dividing the $D_{avg}(p) = [D(p,n-\log_2 l) * (n\text{-}\log_2 l) + \log_2 l] * nc$ over the minimum packet delay, which is simply the transmission delay of all stages, i.e. $n * nc$.

Universal performance factor Upf(p) of an l-layer MIN, can be expressed according to equation (4), and taking into account that $D^{min} = 1$, and $Th^{max} = 2 * l$ by

$$Upf(p) = \sqrt{w_d * (D(p)-1)^2 + w_{th} * \left(\frac{2*l-Th(p)}{Th(p)}\right)^2} * r_p \quad (11)$$

The maximum *normalized throughput* take place when the *multicast ratio* is $m = 1$, and thus the *normalized throughput* at last stage of single-layer segment is also $Th(p,n\text{-}\log_2 l) = 1$. At this case, the second term of equation (9) becomes $2^{1+\log_2 l} = 2 * l$, denoting that each queue of all layers within the non-blocking segment of the MIN forwards 2 packets at each time slot.

V. Simulation and Performance Results

A special-purpose simulator was developed for evaluating the overall network performance of banyan type MINs. This simulator which was written in C++ and includes models for dual priority SEs, it can also handle multicast traffic over multi-layer MINs operating under different configuration schemes. Performance evaluation was conducted using simulation, rather than mathematical modeling, due to the high complexity of the latter [22], stemming from the combination of multicast traffic, multi-priority and multiple buffer positions in SEs. Since partial-multicast transmission was applied, a packet can be serviced either fully at both directions or partially (at one direction only). Consequently, if only one of the successive

buffers is available at the current time slot, the packet is forwarded partially to the corresponding destination link, and a copy remains at the current stage, in order to be later serviced fully. This arrangement is depicted in algorithms 1 and 2, which illustrate the .algorithms for unicast partial forwarding and broadcast forwarding, respectively.

Internally, each SE was modeled by an array of P non-shared buffer queue pairs, where P is the number of priority classes; within each pair, one buffer was dedicated for the upper queuing bank and the other for the lower bank. Buffer operation was based on the FCFS principle. The contention between two packets was resolved randomly, but when a dual 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. Several input parameters such as the *buffer-length,* the *number of input and output ports*, the *number of stages*, the *offered load*, the *multicast ratio,* and the *number of layers* were considered.All simulation experiments were performed at packet level, assuming fixed-length packets transmitted in equal-length time slots, where the slot was the time required to forward one (in the case of unicast) or two (in case of multicast) packet(s) from one stage to the next. In all cases packet contentions were resolved randomly. Algorithm 3 illustrates the overall logic of packet forwarding within the exaf,af"

Metrics such as packet *throughput,* packet *delay,* and *loss probability* were collected. 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 phase to allow the network to reach a steady state, by discarding the data from the first 10^3 network cycles, before initiating metrics collection.

Unicast-Partial-Forwarding (cs_{id}, cl_{id}, nl_{id}, sq_{id}, aq_{id}, pr_{id})
Input: Current stage_id (cs_{id}) ; current and next stage layer_id (cl_{id}, nl_{id}) of send- and accept-queue/s respectively; send-queue_id (sq_{id}) of current stage; accept-queue_id (aq_{id}) of next stage and priority_id (pr_{id}).
Output: Population for send- and accept-queues (Pop) ; total number of serviced and blocked packets for send-queue (Serviced, Blocked) respectively ; total number of packet delay cycles for send-queue (Delay); routing address RA of each buffer position of queue ; partial multicast service indicator for the head of line packet of send-queue (PS).

```
{
  if (Pop[aq_id][cs_id+1][nl_id][pr_id] = B) // blocking state; B is the buffer-size
     Blocked[sq_id][cs_id][cl_id][pr_id] = Blocked[sq_id][cs_id][cl_id][pr_id]+1 ;
  else // unicast-partial forwarding
  {
    Serviced[sq_id][cs_id][cl_id][pr_id] = Serviced[sq_id][cs_id][cl_id][pr_id]+1 ;
    Pop[sq_id][cs_id][cl_id][pr_id] = Pop[sq_id][cs_id][cl_id][pr_id]-1 ;

    Pop[aq_id][cs_id+1][nl_id][pr_id] = Pop[aq_id][cs_id+1][nl_id][pr_id]+1 ;
    RA[aq_id][cs_id+1][nl_id][pr_id][Pop[aq_id][cs_id+1][nl_id][pr_id]] = RA[sq_id][cs_id][cl_id][pr_id][1] ;
    for (bf_id=1; bf_id>=Pop[sq_id][cs_id][cl_id][pr_id]; bf_id++)
       RA[sq_id][cs_id][cl_id][pr_id][bf_id] = RA[sq_id][cs_id][cl_id][pr_id][bf_id+1] ;// RA is the Routing Address of the packet
    // located at (bf_id)^th position of send-queue
    PS[sq_id][cs_id][cl_id][pr_id] = -1 ; // re-initialize the indicator; PS=-1 means the head of line packet is not partially serviced
  }
  Delay[sq_id][cs_id][cl_id][pr_id] = Delay[sq_id][cs_id][cl_id][pr_id]+Pop[sq_id][cs_id][cl_id][pr_id] ;
  return Pop, Serviced, Blocked, Delay, RA, PS ;
}
```

Algorithm 1: Unicast-Partial-Forwarding for multi-layer, multi-priority MINs

Broadcast-Forwarding (cs_{id}, cl_{id}, nl_{id}, sq_{id}, uq_{id}, lq_{id}, pr_{id}, mp)

Input: Current stage_id (cs_{id}) ; current and next stage layer_id (cl_{id}, nl_{id}) of send- and accept-queue/s respectively; send-queue_id (sq_{id}) of current stage ; upper and lower output port queue_id uq_{id}, lq_{id} of next stage accept-queue respectively; priority_id (pr_{id}) and multicast policy (mp).

Output: Population for send- and accept-queues (Pop) ; total number of serviced and blocked packets for send-queue (Serviced, Blocked) respectively ; total number of packet delay cycles for send-queue (Delay); routing address RA of each buffer position of queue ; partial multicast service indicator for the head of line packet of send-queue (PS).

```
{
  if (Pop[uq_id][cs_id+1][nl_id][pr_id] = B) or (Pop[lq_id][cs_id+1][nl_id][pr_id] = B) // blocking state
     Blocked[sq_id][cs_id][cl_id][pr_id] = Blocked[sq_id][cs_id][cl_id][pr_id]+1 ;
  if (Pop[uq_id][cs_id+1][nl_id][pr_id] < B) and (Pop[lq_id][cs_id+1][nl_id][pr_id] < B) // broadcast forwarding
  {
     Serviced[sq_id][cs_id][cl_id][pr_id] = Serviced[sq_id][cs_id][cl_id][pr_id]+1 ;
     Pop[sq_id][cs_id][cl_id][pr_id] = Pop[sq_id][cs_id][cl_id][pr_id]-1 ;
     Pop[uq_id][cs_id+1][nl_id][pr_id] = Pop[uq_id][cs_id+1][nl_id][pr_id]+1 ;
     Pop[lq_id][cs_id+1][nl_id][pr_id] = Pop[lq_id][cs_id+1][nl_id][pr_id]+1 ;
     RA[uq_id][cs_id+1][nl_id][pr_id][Pop[uq_id][cs_id+1][nl_id][pr_id]] = RA[sq_id][cs_id][cl_id][pr_id][1] ;
     RA[lq_id][cs_id+1][nl_id][pr_id][Pop[lq_id][cs_id+1][nl_id][pr_id]] = RA[sq_id][cs_id][cl_id][pr_id][1] ;
     for (bf_id=1; bf_id>=Pop[sq_id][cs_id][cl_id][pr_id]; bf_id++)
         RA[sq_id][cs_id][cl_id][pr_id][bf_id]= RA[sq_id][cs_id][cl_id][pr_id][bf_id+1] ;
         // where RA is the Routing Address of the packet
         // located at (bf_id)^th position of send-queue
     PS[sq_id][cs_id][cl_id][pr_id] = -1 ; // initialize again the indicator
  }
  if ( mp="partial") // partial multicast forwarding is enabled
  {
      if  (Pop[uq_id][cs_id+1][nl_id][pr_id] < B) and (Pop[lq_id][cs_id+1][nl_id][pr_id] = B)
      { // upper port partial multicast service
         Pop[uq_id][cs_id+1][nl_id][pr_id] = Pop[uq_id][cs_id+1][nl_id][pr_id]+1 ;
         RA[uq_id][cs_id+1][nl_id][pr_id][Pop[uq_id][cs_id+1][nl_id][pr_id]] = RA[sq_id][csid][cl_id][pr_id][1] ;
         PS[sq_id][cs_id][cl_id][pr_id] = 0 ;
      }
      if (Pop[uq_id][cs_id+1][nl_id][pr_id] = B) and (Pop[lq_id][cs_id+1][nl_id][pr_id] < B)
      { // lower port partial multicast service
         Pop[lq_{id}][cs_{id}+1][nl_{id}][pr_{id}] = Pop[lq_{id}][cs_{id}+1][nl_{id}][pr_{id}] +1 ;
         RA[lq_id][cs_id+1][nl_id][pr_id][Pop[lq_id][cs_id+1][nl_id][pr_id]] = RA[sq_id][cs_id][cl_id][pr_id][1] ;
         PS[sq_id][cs_id][cl_id][pr_id] = 1 ;
      }
  }
  Delay[sq_id][cs_id][cl_id][pr_id] = Delay[sq_id][cs_id][cl_id][pr_id] + Pop[sq_id][cs_id][cl_id][pr_id] ;
  return Pop, Serviced, Blocked, Delay, RA, PS ;
}
```

Algorithm 2: Broadcast-Forwarding for multi-layer, multi-priority MINs

```
SendQueue-Process (cs_id, cl_id, nl_id, sq_id, mp)
Input: Current stage_id (cs_{id}) ; current and next stage layer_id (cl_id, nl_id) of send- and accept-queue/s respectively ;
send-queue_id (sq_id) of current stage and multicast policy (mp).
{
   processor=0 ;
   for (pr_id=P-1; pr_id>=0; pr_id--) // P is the total number of priorities
      if (Pop[sq_id][cs_id][cl_id][pr_id] >0) and (processor=0)
      // pr_id-class send-queue is not empty and processor is still ready for forwarding
      {
         RA_bit=get_bit(RA[sq_id][cs_id][clid][pr_id][1]) ; // Routing Address (RA)
         MM_bit=get_bit(MM[sq_id][cs_id][cl_id][pr_id][1]) ; // Multicast Mask (MM)
        // get the (cs_id)th bit of (RA) and (MM) of the leading packet
        // of pr_id-class send-queue by a cyclic logical left shift respectively
        if ( (mp="full") and (MM_bit = 1) ) or // broadcast forwarding
          ( (mp="partial")  and (MM_bit = 1)  and  (PS[sq_id][cs_id][cl_id][pr_id] = -1) )
        // where PS=-1 means the head of line packet has no partially serviced
        {
           uq_id = 2 * (sq_id % (N/2) ) ; // upper link for perfect shuffle algorithm
           lq_id = 2 * (sq_id % (N/2) ) +1 ; // lower link for perfect shuffle algorithm
           Broadcast-Forwarding (cs_id, cl_id, nl_id, sq_id, uq_id, lq_id, pr_id, mp) ;
        }
        else // unicast or partial multicast forwarding
        {
          if (RA_bit= 0) or ( (mp="partial")  and  (MM_bit= 1) and  (PS[sq_id][cs_id][cl_id][pr_id]= 1) )
          // upper port forwarding
                aq_id = 2 * (sq_id % (N/2) ) ; // link for perfect shuffle algorithm
          else if (RA_bit=1) or ( (mp="partial")  and (MM_bit=1) and (PS[sq_id][cs_id][cl_id][pr_id]=2) )
          // lower port forwarding
                aq_id = 2 * (sq_id % (N/2) ) +1 ; // link for perfect shuffle algorithm
          // where PS=1, 2 means packet has been serviced partially at lower or upper port direction, respectively
          Unicast-Partial-Forwarding (cs_id, cl_id, nl_id, sq_id, aq_id, pr_id, mp) ;
        }
        processor=1 ;
      }
}
```

Algorithm 3: SendQueue-Process for multi-layer, multi-priority MINs

A. Simulator validation

Single-layer MINs were modeled for validating our simulation experiments. All results obtained from this simulation were compared against those reported in other works which are considered the most accurate ones under both unicast and multicast traffic. Thus, for the case of unicast traffic, whereas m=0, we noticed that all simulation

experiments (fig. 3, curve SP-B1-M:0) were in close agreement with the results reported in [17] (fig. 2), and -notably- as Theimer's model [20], which is considered to be the most accurate one.

All SP-B*X*-M:*Y* curves at subsequent diagrams denote the performance of a single-priority, 6-stage MIN whose SEs in the single-layer segment have *buffer size* equal to *X* and operating with *multicast ratio m* equal to *Y*%. Similarly, HP-B*X*-M:*Y* and LP-B*X*-M:*Y* curves depict the performance of high and low priority traffic of a dual priority MIN respectively.

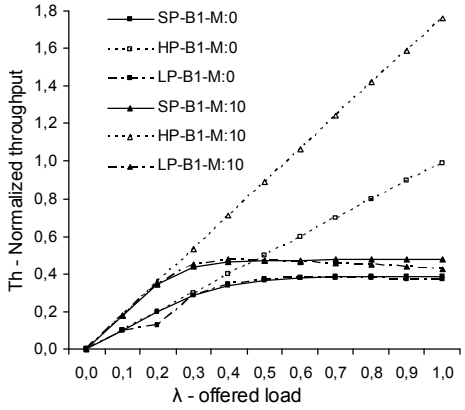

Figure 3. Normalized throughput of single-layer MINs vs. offered load (m=0.10)

Fig. 4 represents the *relative normalized throughput* of a dual vs. single priority mechanism for a single-buffered, 6-stage, single-layer MIN at the case of using partial multicasting policy, in the case m=0.5. We compared our measurements (fig.3 curve SP-B1-M:50) against those obtained from Tutsch's Model reported in [10] (fig.8 solid curve in the referenced paper), when all possible combinations of destination addresses for each packet entering the network were equally distributed, and we have found that both results are in close agreement (*normalized throughput* is about 75% in both papers).

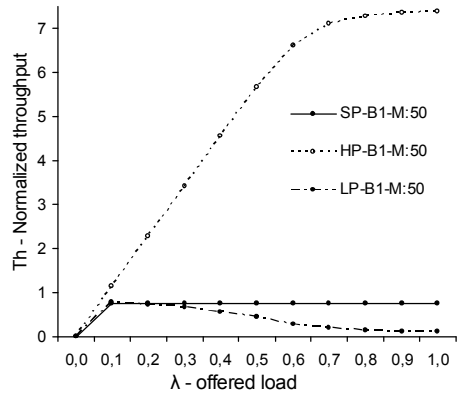

Figure 4. Normalized throughput of single-layer MINs vs. offered load (m=0.50)

B. Multicasting on Dual-priority, Single-layer MINs

The introduction of a dual priority scheme in a single-layer MIN has a significant impact on the quality of service offered to packets having different priorities. Figures 4 and 5 depict the normalized throughput and normalized delay of high- and low-priority packets, in an "heavy multicasting" scenario [the probability that a packet is unicast is equal to $(1-0.5)^6$=0.016, i.e. 98.4% of the packets are multicast to at least two destinations); the respective metrics for a single-priority scheme are also included for comparison.

Due to the heavy multicasting, the network is quickly saturated and we can observe that in the single priority setup, peak throughput is reached at λ=0.1, and it remains constant thereafter. In the dual priority setup, high priority packets (recall that these correspond to 10% of the overall traffic in the described experiments) are serviced at almost optimal QoS for loads $\lambda \leq 0.7$. Low priority packets exhibit a throughput drop, which is small for loads $\lambda \leq 0.3$ and *tolerable* for loads $\lambda \leq 0.5$, while for higher loads the performance drop is considerable.

Regarding the delay, we can observe in fig. 5 that for loads $\lambda \leq 0.4$ high-priority packets traverse the network with almost no blockings, while for loads $\lambda \geq 0.6$ the effect of blockings on the delay of high-priority packets is noticeable. Low priority packets have a high delay even at small loads, and beyond the point of λ=0.5 the delay rises sharply.

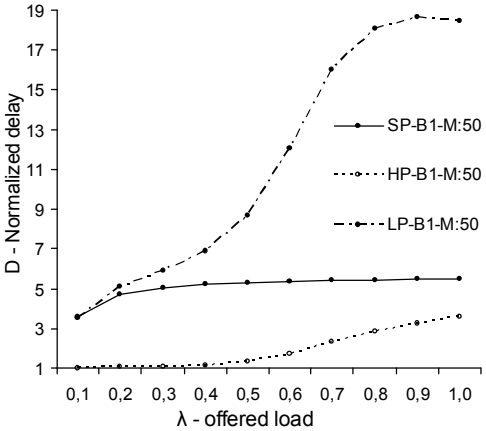

Figure 5. Normalized delay of single-layer MINs vs. offered load (m=0.50)

Respectively, figures 3 and 6 illustrate the *normalized throughput* and *delay* of high- and low-priority packets in a unicast (*m=0*) and in a moderate multicasting scenario (*m=0.1*, which in a 6-stage MIN results to approximately half the packets being multicasted to at least two destinations).

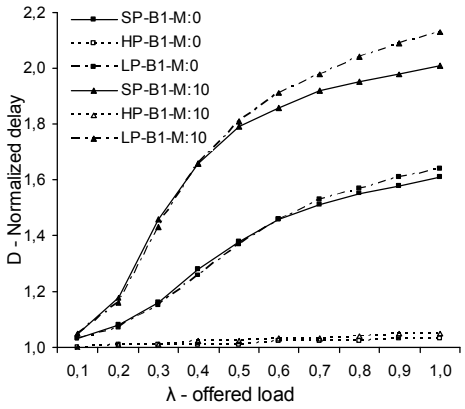

Figure 6. Normalized delay of single-layer MINs vs. offered load (m=0.10)

Under both scenarios, high-priority packets receive an almost optimal quality of service, both in terms of *throughput* and *delay*, under all loads, while low-priority packets sustain some observable performance drop in the moderate multicasting scenario only; throughput appears to drop –as compared to the single priority setup- for very high overall loads ($\lambda \geq 0.8$), while increase in delay becomes apparent at a somewhat smaller load ($\lambda \geq 0.7$). We should note here that the introduction of the dual priority scheme, effectively increases the buffer capacity of SEs, since separate buffer queues are dedicated to packets of each priority, and this is the reason why both the *overall throughput* and *delay* of the network increases (for a discussion on the effects of SE *buffer size* on overall MIN performance, the interested reader is referred to [21]).

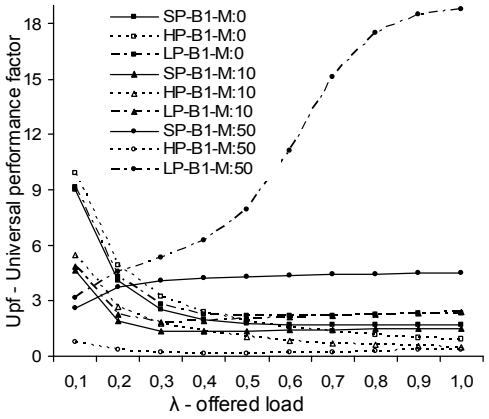

Figure 7. Universal performance factor of single-layer MINs vs. offered load

Figure 7 depicts the MIN *universal performance factor* for the three above scenarios (unicast, moderate multicast and heavy multicast), for both the dual- and the single-priority setups. The findings reaffirm that high priority packets enjoy an almost optimal quality of service, even at high loads. In the unicast and moderate multicast scenario, low-priority packets reach their peak quality of service for λ=0.4, and this slightly drops for higher loads; in the heavy multicast scenario, the quality of service offered to low-priority packets is initially tolerable, but deteriorates sharply as load increases.

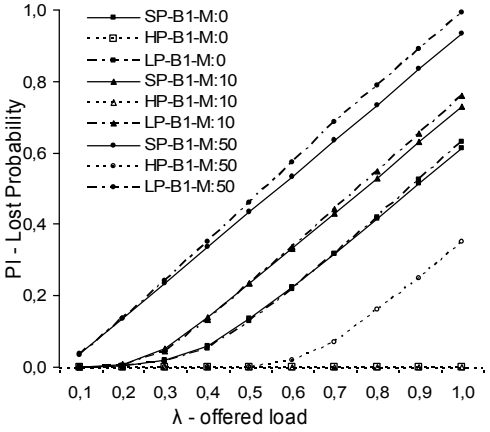

Figure 8. Packet loss probability of single-layer MINs vs. offered load

Packet loss probability is another metric that should be taken into account, in order to assess the overall MIN operation. As we can see in figure 8, under the unicast and moderate multicast scenario, high-priority packets are never or rarely dropped; under the heavy multicast scenario, however, for loads λ>0.5 the network begins to drop packets, since there is not enough capacity to route all packets to their destinations (recall packets can be cloned as they traverse the network, thus the overall number of packets increases). The probability that a low-priority packet is lost is comparable to that of packet loss in a single priority setup, being observably higher under the heavy multicast scenario for loads $\lambda \geq 0.6$. Note that under this scenario and in the extreme case that λ=1, the *packet loss probability* is close to 1.

C. Multicasting on Dual-priority Multi-layer MINs

In this section, we present our findings for a 6-stage MIN where the number of layers at the last stage l is equal to 4, i.e. the first four stages are single-layer and multiple layers are only used at the last two stages, in an attempt to balance between MIN performance and cost. For the first 4 stages, single- and double-buffered SEs are considered, whereas at the last two stages (which are non-blocking), single-buffered SEs are used, as the absence of blockings removes the need for larger buffers.

Figure 9 illustrates the *normalized throughput* for the heavy multicast scenario, considering SEs of buffer size 1 and 2; single-priority metrics are also included for reference. Under both scenarios, high-priority packets are served optimally, while low-priority packets enjoy a better service when buffer size b is equal to 2; this is consistent with findings of other works (e.g. [21]), reporting that double buffers lead to increased *throughput*. Note that low-

priority packets appear to enjoy better throughput *even compared to the single-priority setup* (except for loads $\lambda \geq 0.8$ in the single-buffer configuration), and this is again owing to the additional buffers available in the SEs, due to the fact that different buffer queues are dedicated to packets of different priorities. The overall throughput for high-priority packets has increased by more than 50% in the high load area ($\lambda > 0.7$) and the *throughput* of low-priority packets has effectively quadrupled, as compared to the single-layer MIN at almost all loads. Taking into account that introducing two stages of multi-layered SEs has lead to increasing the routing capacity in the last two stages by four, we can conclude that this increase is exploited to an almost full extent.

Figure 10 depicts the *normalized throughput* for high- and low-priority packets under the moderate multicast scenario; single-priority metrics are also included for reference. High-priority packets are again served optimally (but no considerable difference can be seen against the single-layer case), while for low-priority packets the performance gains are approximately 100% when *buffer size* is set to 2, and approximately 67% when *buffer size* is compared to 1 (both comparisons are against the single-layer, single-buffer configuration), thus *buffer size* plays an important role in this setup. Note also that the network reaches its peak *throughout* regarding low-priority packets at load λ=0.6, as opposed to the single-layer case where peak performance is attained at load λ=0.3, indicating that the additional bandwidth offered by the multi-layer SEs is exploited to some good extent.

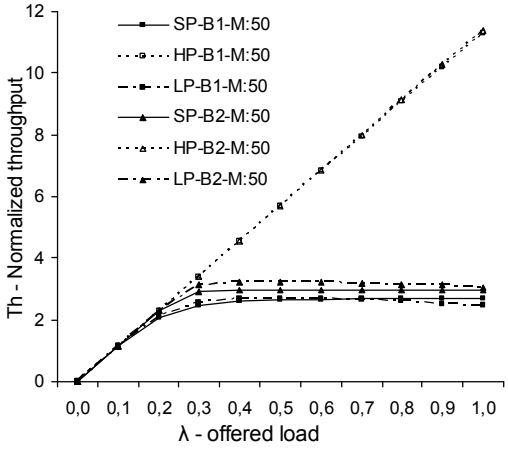

Figure 9. Normalized throughput of multi-layer MINs vs. offered load (m=0.50)

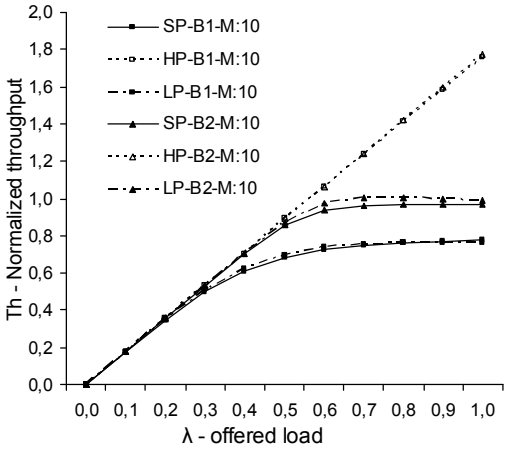

Figure 10. Normalized throughput of multi-layer MINs vs. offered load (m=0.10)

Figures 11 and 12 show *the normalized delay* for the heavy- and medium-multicasting scenarios, respectively, considering both *buffer sizes* (1 and 2) and both priority schemes (dual- and single-priority). The *normalized delay* for high-priority packets is close to the optimal in all cases. Expectedly, when the double-buffered configuration is considered, delays are increased for low-priority packets and packets in the single-priority setup; the point beyond which the increment is considerable differs across the different scenarios, being located at λ=0.3 for heavy multicasting, while for medium multicasting it has been shifted to λ=0.5. In the heavy multicasting scenario, for loads $\lambda \leq 0.3$ low-priority packets have delays comparable to those of packets in the single-priority setup, indicating that in this load range, the network has enough capacity to offer elevated quality of service to high-priority packets, without harming the quality of service offered to low-priority packets. In the medium multicasting scenario, the deviation the *delay metrics* for these classes of packets presents some observable deviation for loads $\lambda \geq 0.5$.

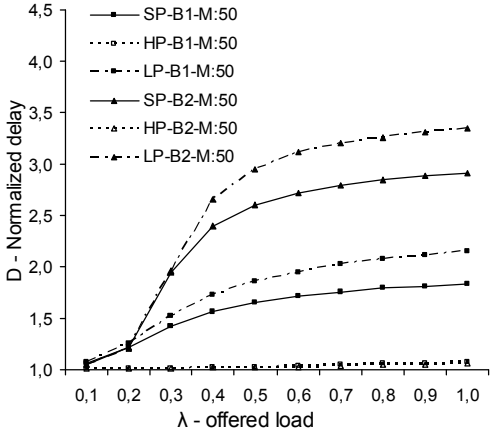

Figure 11. Normalized delay of multi-layer MINs vs. offered load (m=0.50)

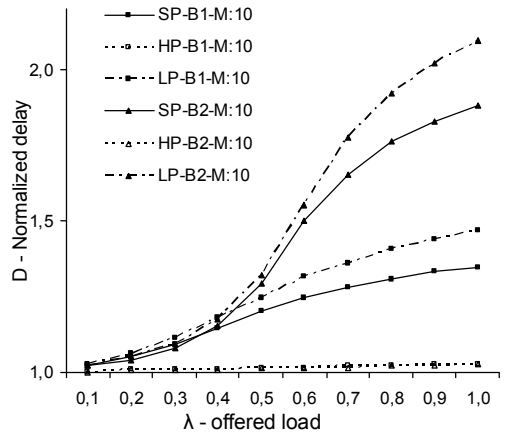

Figure 12. Normalized delay of multi-layer MINs vs. offered load (m=0.10)

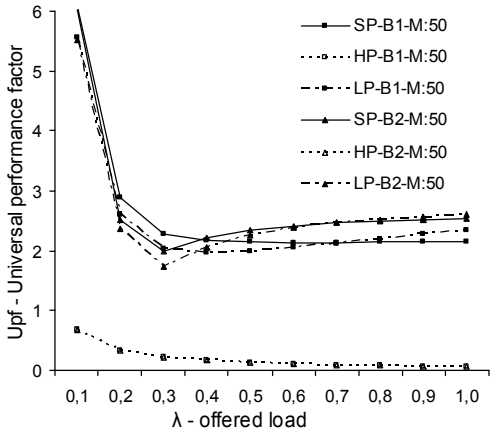

Figure 13. Universal performance factor of multi-layer MINs vs. offered load (m=0.50)

Figure 13 illustrates the *universal performance factor* for the heavy multicasting scenario considering both *buffer sizes* (1 and 2) and both priority schemes. We can observe that the *UPF* for high-priority packets continuously increases with the offered load under all configurations, indicating that the network has amble capacity to service optimally the increasing number of incoming high-priority packets. Regarding low-priority packets and packets in the single-priority setup, we can observe that initially the *UPF* improve in all setups, as more packets enter the network and therefore *normalized throughput* increases. In the heavy multicasting scenario and for *buffer size b*=2, the optimal point is reached at λ=0.3, while beyond that point the sharp deterioration in the delay dominates over the small increments in the *throughput*, and thus the *UPF* appears to drop from that point onwards. In the same curves (SP-B2-M:50 and LP-B2-M:50), we can notice that for loads $\lambda \leq 0.6$, the low-priority packets have a better overall quality of service as compared to the packets in the single priority setup: this is owing to the throughput gains obtained by the introduction of the additional buffer queues, required for implementing the priority mechanism. For loads, however, $\lambda \geq 0.8$, the increments in delay diminish these gains, and the overall QoS for low-priority packets appears smaller than the respective QoS of packets in the single priority setup. Analogous remarks hold for the single-buffered configuration, with the overall performance drop point being located at λ=0.5, and the point beyond which low priority packets begin to be served worse than packets in the single priority setup being located at λ=0.7.

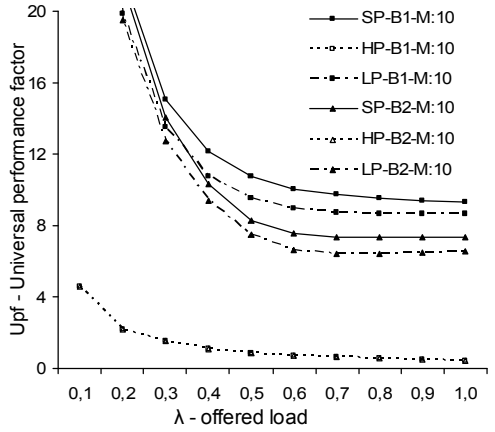

Figure 14. Universal performance factor of multi-layer MINs vs. offered load (m=0.10)

Figure 14 illustrates the UPF for the medium multicasting scenario. High-priority packets are again served optimally, with the UPF exhibiting a sharper improvement as the load increases: this is due to the fact that for low loads, the network capacity is underutilized and hence the improvement potential is high. Regarding low-priority packets and packets in the single-priority scheme, we can notice here that double-buffered setups are consistently better than their single-buffer counterparts at all loads, indicating that the throughput gains obtained due to the introduction of additional buffer queues dominate over the deteriorations in the delay. Beyond the point of λ=0.7, only minor improvements can be observed on the UPF for these cases, indicating that at this point the network has been saturated.

Finally, figure 15 illustrates the **packet loss probability** for the heavy multicast scenario. We can observe that while high-priority packets are never lost, low-priority packets and packets in the single-priority setup can be lost at particularly light loads ($\lambda \geq 0.2$ for single-buffer

configurations and $\lambda \geq 0.3$ for double-buffered configurations). As anticipated, double-buffer configurations achieve a lower *packet loss probability*, since in these setups it is more likely that a buffer space is available to accommodate an incoming packet. In this diagram, it is worth noting that beyond the point of $\lambda \geq 0.6$, the packet loss probability of the *single-buffered dual-priority* setup is less than the packet loss probability of the *double-buffered single-priority* setup. This stems from the availability of the extra buffer queues in the dual priority setup, which –beyond that point– are utilized at maximum capacity.

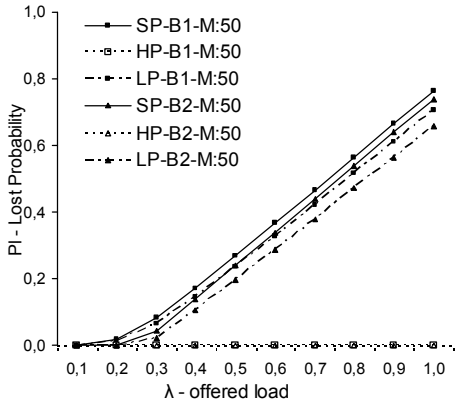

Figure 15. Packet loss probability of multi-layer MINs vs. offered load (m=0.50)

D. Multicasting at 2-class Priority Multi-layer Segment

In this section we present our findings for a special operational mode of the multi-layer MIN, in which multicasting occurs only at the last $\log_2 l+1$ stages, i.e. packet cloning due to multicasting occurs only in the non-blocking segment. This mode of operation may be applied, for example, to cases of interconnected LANs, where multicasting/broadcasting can be performed within the limits of a single LAN but traffic across distinct LANs is always unicast. As an example, setting l=16 in a (64x64) MIN produces a configuration that can serve two interconnected LANs of 32 nodes each. A MIN in this mode combines both the LAN switch and the network trunk functionalities.

In the diagrams below, performance metrics are illustrated for different values of m (0.1, 0.5) and for l=4, thus multicasting occurs only in the last 3 stages. Since these stages are non-blocking, both *delay* and *loss probability* are not affected by the value of m and are only related to the *offered load* λ and the *buffer size* of the SEs in the single-layer segment.

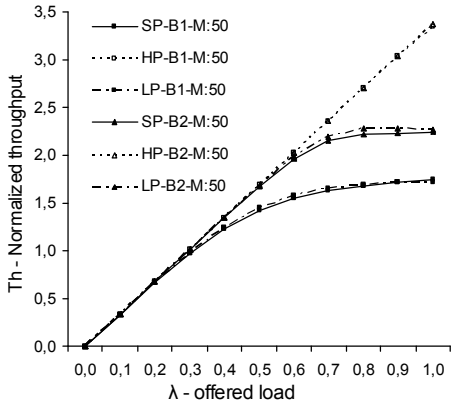

Figure 16. Normalized throughput of multi-layer MINs vs. offered load (m=0.50)

Therefore, variable m has been eliminated from diagrams depicting *packet delay* (fig. 18) and *loss probability* (fig. 21), and these performance factors are analyzed with respect only to the offered load λ and the *buffer size*. For both these performance factors, we can comment that their absolute values remain low, and are even lower than the corresponding metrics collected for unicast traffic in single-layer MINs (fig. 6 and fig. 8, respectively). This is owing (a) to the availability of extra buffer spaces due to the implementation of the dual priority scheme and (b) due to the fact that the probability of blockings in the last two stages drops to zero, whereas in fig. 6 and fig. 8 this does not hold. We can notice however that the *normalized throughput* in the heavy multicast scenario (figure 16) differs from the *normalized throughput* in the medium multicast scenario (figure 17): this is natural, since in the former case, more packet clonings occur (due to increased multicasting), thus more packets reach a destination port within any time unit.

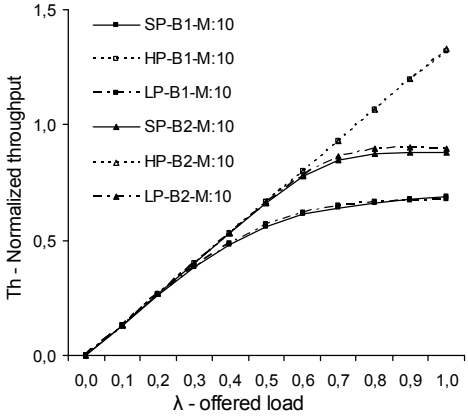

Figure 17. Normalized throughput of multi-layer MINs vs. offered load (m=0.10)

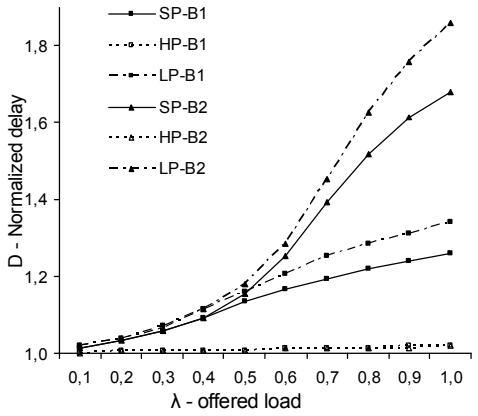

Figure 18. Normalized delay of multi-layer MINs vs. offered load

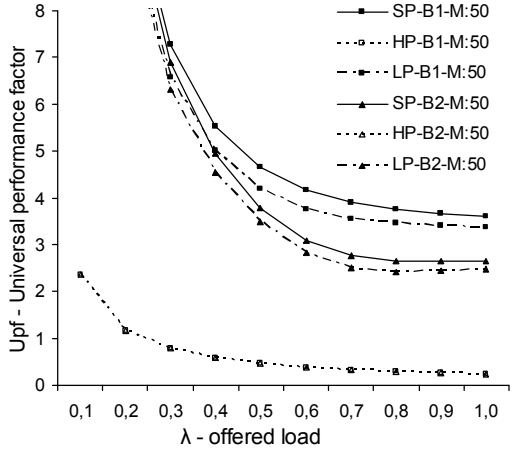

Figure 19. Universal performance factor of multi-layer MINs vs. offered load (m=0.50)

Figure 19 illustrates the *universal performance factor* for the heavy-multicast scenario. We can notice here a consistent improvement of the *UPF*, until the point that the single-layer segment is saturated (λ=0.7). Beyond that point, small increments in *throughput* are counterbalanced with the increments in *delay*. Similar conclusions can be drawn from figure 20, which illustrates the *UPF* for the medium-multicast scenario. We can notice here that the absolute values of *UPF* are considerably higher (thus the network performance is considered worse), mainly owing to the reduced values of *throughput* (less packets traverse the network due to reduced multicasting). Again, double-buffered setups appear to have a performance edge over their single-buffered counterparts, since they are able to attain higher *throughput values*, and the respective increase in the *delay*, while existent, is not sufficient to cancel this advantage.

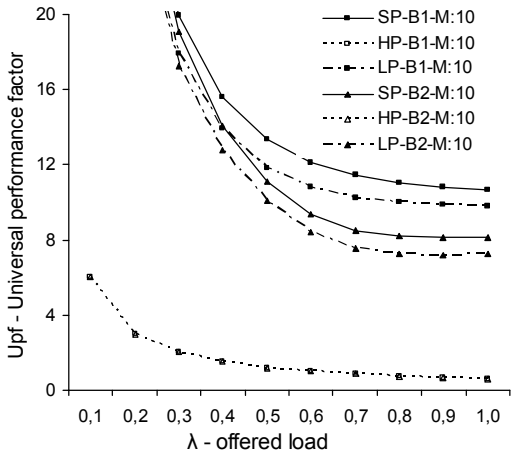

Figure 20. Universal performance factor of multi-layer MINs vs. offered load (m=0.10)

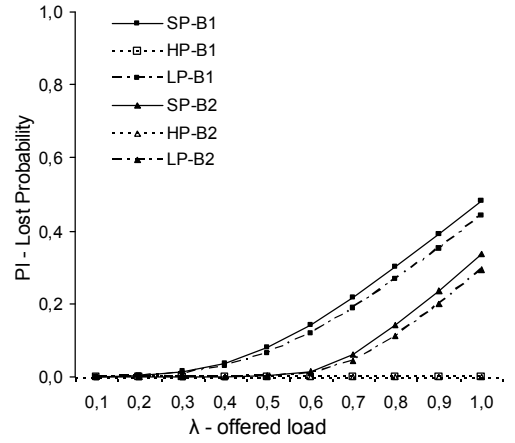

Figure 21. Loss probability of multi-layer MINs vs. offered load

Finally, fig. 21 illustrates the *packet loss probability* against the *offered load*, both for the dual- and single-priority scenario. Again, high-priority packets are not lost under any circumstances, while double-buffed setups expectedly achieve a smaller *loss probability*, since it is more likely that an incoming packet can find available buffer space. Dual priority configurations exhibit also smaller *packet loss probability* compared to their single-priority counterparts, due to the existence of additional queues.

VI. CONCLUSIONS

Multistage Interconnection Network technology is a prominent approach for implementing NGNs, having an appealing cost/performance ratio and high performance. Multicasting however, which is a core requirement for NGNs, has been found to significantly degrade MIN performance, and multi-layer MINs have been introduced to cope with traffic shapes involving multicasting. In this

paper, we extensively study the performance of multi-layer MINs operating under various *overall input loads* and *multicast packet ratios*, considering also a dual-priority scheme. We have additionally taken into account different *buffer size* configurations for SEs, and more specifically *buffer sizes* equal to 1 and 2, which are proven to be the most efficient ones. For all these configurations, we have drawn conclusions regarding the *network throughput*, the *packet delay* and the *packet loss probability*, and we have also computed the *Universal Performance Factor*, a metric combining *throughput* and *delay*. The findings of this performance evaluation can be used by network designers for drawing optimal configurations while setting up MINs, so as to best meet the performance and cost requirements under the anticipated traffic load and quality of service specifications. The presented results also facilitate performance prediction for multi-layer MINs before actual network implementation, through which deployment cost and rollout time can be minimized.

Future work will focus on examining other load configurations, including hotspot and burst loads, as well as performance evaluation under multiple priority schemes (i.e. more than two classes of priorities). Variations in the multi-layer configuration, including cases where the number of layers in the multi-layer segment increases by a factor less than two in each subsequent stage, in an attempt to reduce the overall MIN cost will be studied.

REFERENCES

[1] OECD. Convergence and Next Generation Networks. 2007. http://www.oecd.org/dataoecd/25/11/40761101.pdf

[2] G. F. Goke, G.J. Lipovski. "Banyan Networks for Partitioning Multiprocessor Systems" Procs. of 1st Annual Symposium on Computer Architecture, pp. 21-28, 1973.

[3] J.H. Patel. "Processor-memory interconnections for mutliprocessors", Procs. of 6th Annual Symposium on Computer Architecture. New York, pp. 168-177, 1979.

[4] D. A. Lawrie. "Access and alignment of data in an array processor", IEEE Transactions on Computers, C-24(12):11451155, Dec. 1975.

[5] G. B. Adams and H. J. Siegel, "The extra stage cube: A fault-tolerant interconnection network for supersystems", IEEE Trans. on Computers, 31(4)5, pp. 443-454, May 1982.

[6] Cisco Systems. Next generation networks and the CISCO carrier routing system, http://newsroom.cisco.com/dlls/2004/next_generation_networks_and_the_cisco_carrier_routing_system_overview.pdf (2004).

[7] CISCO Systems. Service Providers Worldwide Driving Video/IPTV with Cisco IP NGN. 2005. http://newsroom.cisco.com/dlls/2005/prod_090905b.html

[8] International Telecommunication Union (ITU). Draft Recommendation Y.NGN-FRA R2, "Functional requirements and architecture of the NGN of release 2," SG13, Sep. 2007

[9] International Telecommunication Union (ITU). ITU-T Draft Recommendation Y.ngn-mcastsf, "NGN Multicast Service Framework" SG13, Sep. 2007

[10] D. Tutsch, G.Hommel. "Comparing Switch and Buffer Sizes of Multistage Interconnection Networks in Case of Multicast Traffic", Procs. of the High Performance Computing Symposium, (HPC 2002); San Diego, SCS, pp. 300-305, 2002.

[11] D. Tutsch, M. Hendler, and G. Hommel, "Multicast Performance of Multistage Interconnection Networks with Shared Buffering", Proceedings of ICN 2001, LNCS 2093, pp. 478–487, 2001.

[12] Jaehyung Park and Hyunsoo Yoon. "Cost-effective algorithms for multicast connection in ATM switches based on self-routing multistage networks", Computer Communications, vol. 21, pp. 54–64, 1998.

[13] Rajeev Sivaram, Dhabaleswar K. Panda, and Craig B. Stunkel. "Efficient broadcast and multicast on multistage interconnection networks using multiport encoding", IEEE Transaction on Parallel and Distributed Systems, vol. 9(10), pp. 1004–1028, October 1998.

[14] Neeraj K. Sharma. "Review of recent shared memory based ATM switches" Computer Communications, vol. 22, pp. 297–316, 1999.

[15] D. Tutsch and G. Hommel. "Multilayer Multistage Interconnection Networks", Proceedings of 2003 Design, Analysis, and Simulation of Distributed Systems (DASD'03). Orlando, USA, pp. 155-162, 2003.

[16] G. Shabati, I. Cidon, and M. Sidi, "Two priority buffered multistage interconnection networks", Journal of High Speed Networks, pp.131–155, 2006.

[17] D.C. Vasiliadis, G.E. Rizos, C. Vassilakis, and E.Glavas. "Performance evaluation of two-priority network schema for single-buffered Delta Network", Procs. of IEEE PIMRC' 07, Sep.2007.

[18] D. Tutsch and G. Hommel. "Performance of buffered multistage interconnection networks in case of packet multicasting". Proceedings of Advances in Parallel and Distributed Computing, 19-21, pp. 50-57, Mar 1997.

[19] D.C. Vasiliadis, G.E. Rizos, C. Vassilakis, and E.Glavas. "Performance Evaluation of Multicast Routing over Multilayer Multistage Interconnection Network", Procs. Of AICT' 09, IEEE press, 2009.

[20] T.H. Theimer, E. P. Rathgeb and M.N. Huber. "Performance Analysis of Buffered Banyan Networks", IEEE Transactions on Communications, vol. 39, no. 2, pp. 269-277, 1991.

[21] D.C. Vasiliadis, G.E. Rizos, and C. Vassilakis. "Performance Analysis of blocking Banyan Swithces", Procs. of CISSE 06, December, 2006.

[22] J. Garofalakis, and E. Stergiou "An analytical performance model for multistage interconnection networks with blocking", Procs. of CNSR 2008, May 2008