# *Performance Analysis of Multi-Layered Multi-Priority Assymetric-Sized Delta Networks*

D. C. Vasiliadis[a,b] , G. E. Rizos[a,b] , C. Vassilakis[a]
[a]Department of Computer Science and Technology
Faculty of Sciences and Technology
University of Peloponnese
GR-221 00 Tripolis Greece
[b]Network Operations Center
Technological Educational Institute of Epirus,
Arta, GR- 471 00 Greece

***Abstract*——In this paper the performance of multi-layered asymmetric-sized finite-buffered Delta Networks supporting multi-class routing traffic is presented and analyzed in the uniform traffic conditions under various loads using simulations. The rationale behind introducing asymmetric-sized buffered systems is to have a better exploitation of available buffer spaces, while the implementation of multi-layered architecture is applied in order to further improve the overall performance of network. The findings of this performance evaluation can be used by network designers for drawing optimal configurations while setting up the network, so as to best meet the performance and cost requirements under the anticipated traffic load and quality of service specifications.**

***Keywords-Multistage Interconnection Networks, Delta Networks, Banyan Switches, Packet Switching, Multi-Priority Networks, Performance Analysis.***

## 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 [10]. Regarding the network infrastructure, this convergence requires the use of packet-switched equipment that can provide communications with low-latency, high-throughput and QoS-awareness. Multistage Interconnection Networks (MINs) have proved to be an infrastructure that does provide the above-listed characteristics.

MIN technology, having the potential to concurrently route multiple communication tasks and exhibiting very low cost/performance ratio, is widely used for the implementation of Next Generation Networks. MINs are distinguished into two classes: the first class has the Banyan property [7] with its most prominent representatives being Delta Networks [11], Omega Networks [8], and Generalized Cube Networks [1]; the second category includes MINs not having the Banyan property, such as Augmented and CLOS MINs. Among the two classes, the first one is more widely used, since non-Banyan MINs are generally more expensive and complex.

The advantages of MINs have been recognized by the industry too: amongst others, Cisco has built its new CRS-1 router [3, 4] as a multistage switch fabric. The switch fabric that provides the communications path between line cards is 3-stage, self-routed architecture.

The importance of the communication infrastructure in both parallel and distributed systems' performance is of particular importance and therefore much research has targeted the evaluation of the performance of the communication infrastructure. To this end, various methods have been employed, including Markov chains, queuing theory, Petri nets and simulation experiments.

Queuing systems, and in particular single priority ones, have been used to study the *throughput* and *delay* of MINs in a number of articles, such as [5, 6, 15], which consider SEs having a single input buffer. Papers such as [9] extend the above works by considering finite-buffered MINs.

Nowadays, the applications running over the Internet and over enterprise IP networks are quite diverse. Among the applications we can identify interactive ones (e.g. telnet, and instant messaging), bulk data transfer-oriented applications (e.g. ftp, and P2P file downloads), corporate (e.g. database transactions), and real-time applications (e.g. voice, and video streaming). The communication requirements posed by these applications vary greatly regarding the quality of service aspects: for instance interactive applications require minimal delays, bulk data transfer applications need high throughput, while streaming applications require small (or at least bounded) jitter. An important means for expressing these requirements to the network layer is *packet priorities*, which are specified by the applications producing the packets. Notably, provisions for packet priorities can be found in protocol specifications, such as the case of TCP out-of-band/expedited data, which are normally prioritized against normal connection data [14].

In order to accommodate packet prioritization, dual priority queuing systems have been introduced in MINs, providing the ability to offer different QoS parameters to packets that have different priorities. Dual-priority MINs employ SEs with two buffer position, where one buffer position is dedicated to low priority packets and one buffer position is assigned to high priority traffic. The performance of dual priority MINs has been investigated insofar in a limited number of works, including [13, 20].

In corporate environments, however, hosting a multitude of applications, two priorities may not be sufficient to express the diversity of application-level requirements to the network layer. [12] argues that besides the inherently different QoS

requirements of different types of applications, priority classification is further refined by (a) the different relative importance of different applications to the enterprise (e.g. database transactions may be considered critical and therefore high priority, while traffic associated with browsing external web sites is generally less important) and (b) the desire to optimize the usage of their existing network infrastructures under finite capacity and cost constraints, while ensuring good performance for important applications. Therefore, it is important that the underlying communication infrastructure supports multiple priorities, to naturally map the application-level priority classes to priority levels within the communication infrastructure.

In this paper we examine MINs that natively support multi-class routing traffic using double-buffered queues in order to offer better QoS, while providing in parallel better overall network performance. Contrary to the majority of the works, which use equal buffer queue sizes for all priority classes [19, 20], in this paper we considered *asymmetric-sized* buffered SEs, i.e. the number of buffer positions dedicated to each packet priority class within each SE is (potentially) different. The motivation for this differentiation is the observation that -typically- normal priority packets outnumber their high-priority counterparts and therefore analogous provisions must be made in terms of buffer spaces. We employ a variation of double-buffered SEs that uses asymmetric buffer sizes [21] for packets of different priorities, aiming to better exploit the network hardware resources and capacity. We also extend previous studies in the area of performance evaluation of MINs (e.g. [13, 20, 21]) by including multi-layer MINs [18, 22], attempting to increase network capacity so as to better service lower-priority packets, which may not be adequately serviced by a single-layer MIN [22].

The remainder of this paper is organized as follows: in section 2 we briefly analyze a Delta Network that natively supports multi-class routing traffic. Subsequently, in section 3 we introduce the performance criteria and parameters related to this network. Section 4 presents the results of our performance analysis, which has been conducted through simulation experiments, while section 5 provides the concluding remarks

## II. ANALYSIS OF MULTI-LAYERED MULTI-PRIORITY DELTA NETWORKS

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 [7] 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.

According to figure 1 each SE is modelled by as an array of $p$ non-shared buffer queue pairs, where $p$ is the number of distinct priority classes supported by the network, with the $i^{th}$ element of the array being dedicated to packets of priority class $i$. Within each pair, one buffer queue 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 thus 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. 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 a SE will be forwarded to the next stage. The priority of each packet is indicated through the appropriate priority bits in the packet header.

An ($N$ X $N$) MIN can be constructed by $n$=$\log_c N$ stages of ($c$x$c$) SEs, where $c$ is the degree of the SEs. 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.

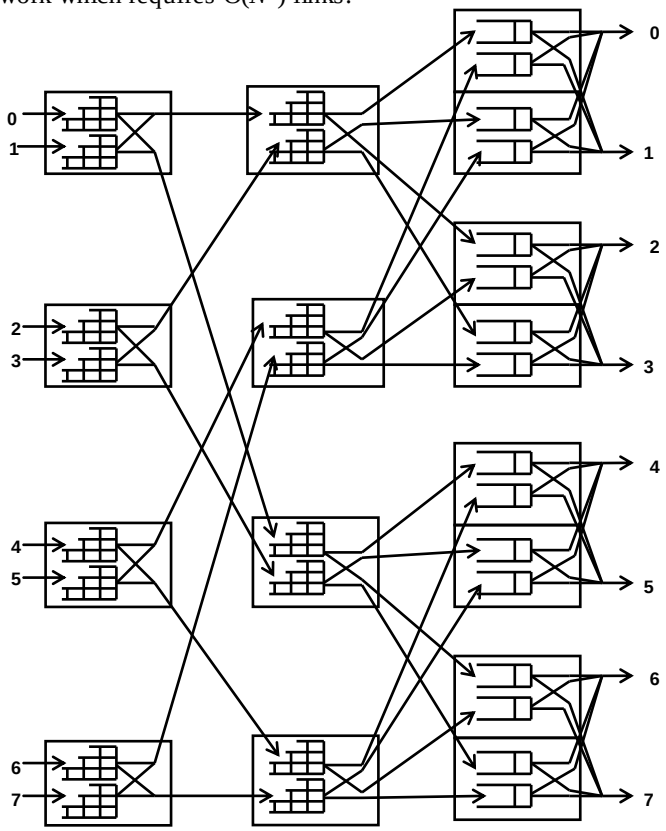


FIGURE 1: An 8 X 8 Delta Network with an asymetric segment (first two stages) and a multi-layer segment (last stage)

A typical configuration of an 8 X 8 Delta Network, a widely used class of Banyan MINs, is depicted in figure 1 and outlined below. This network class was proposed by Patel [6] and combines benefits of Omega [7] and Generalized Cube

Networks [8] (destination routing, partitioning and expandability).

In this paper we extend previous studies by considering multi-layer MINs. Figure 1 represents an example (8 x 8) multi-layer MIN, which employs multiple layers only at the final stage. Thus, this network consists of two segments, an initial single-layer one and a subsequent multi-layer one (with 2 layers). Generally, absence of contention 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 $2*l_i$ (i.e. stage $i$+1 contains twice as many SEs as stage $i$). 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 [18], 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. Therefore, SEs in the last stage has only one buffer position, per input link, to store the packet currently processed; no more buffer positions are necessary, since no blocking can occur in the multi-layer stages.

The rationale behind choosing such an architecture is to have switching elements and more paths (and therefore more routing power) available at the final stages of the MIN. This attribute is also very useful when other load traffic types are applied [22] e.g. hotspot traffic, where the bottlenecks at last stages is very severe.

We also note that the addition of multiple layers in the final stages effectively creates multiple paths between sources and destinations; therefore the MIN as a whole does not have the Banyan property. The MINs considered in this study retain the Banyan property within the initial, single-layer segment, while this property is dropped in the final, multi-layer one.

In our study we used a Delta Network that is assumed to operate under the following conditions:

- The MIN operates in a slotted time model [2]. In each time slot two phases take place. In the first phase, control information passes via the network from the last stage to the first one. In the second phase, packets flow from the first stage towards the last, in accordance to the flow control information.
- At each input of every switch of the MIN only one packet can be accepted within a time slot which is marked by a priority tag, and it is routed to the appropriate class queue. The domain value for this special priority tag in the header field of the packet determines its $i$-class priority, where $i$=1..$p$.
- 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.
- An $i$-class priority packet arriving at the first stage is discarded if the corresponding $i$-class priority buffer of the SE is full, where $i$=1..$p$.
- A backpressure blocking mechanism is used, according to which an $i$-class priority packet is blocked at a stage if the destination of the corresponding $i$-class priority buffer at the next stage is full, where $i$=1..$p$.
- All $i$-class priority packets are uniformly distributed across all the destinations and each $i$-class priority queue uses a FIFO policy for all output ports, where $i$=1..$p$.
- The conflict resolution procedure of a multi-class priority MIN takes into account the packet priority: if one of the received packets is of higher-priority and the other is of lower priority, the higher-priority packet will be maintained and the lower-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. It suffices for the SE to read the incoming packets' headers 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.

## III. Performance Evaluation Methodology

In order to evaluate the performance of multi-priority ($N$X$N$) MIN the following metrics are used. Let $Th_{avg}$ and $D_{avg}$ be the *average throughput* (*bandwidth*) and *average delay* of a MIN respectively.

***Normalized throughput Th*** [26] is the ratio of the *average throughput* $Th_{avg}$ to number of network outputs $N$. Formally, $Th$ can be expressed by

$$Th = \frac{Th_{avg}}{N} \qquad (1)$$

and reflects how effectively network capacity is used.

***Relative normalized throughput RTh(i) of i-class priority traffic***, where $i$=1..$p$ is the *normalized throughput Th(i)* of $i$-class priority packets divided by the corresponding-class *offered load* $\lambda(i)$ of such packets.

$$RTh(i) = \frac{Th(i)}{\lambda(i)} \qquad (2)$$

The definition of relative normalized throughput RTh(i) effectively extends the definition of normalized throughput in [26] to consider the different priority classes.

***Normalized packet delay D(i) of i-class priority traffic***, where $i$=1..$p$ is the ratio of the $D_{avg}(i)$ to the minimum packet delay which is simply the transmission delay $n*nc$ (i.e. zero queuing delay) , where $n$=$\log_2 N$ is the number of intermediate stages and $nc$ is the network cycle. Formally, $D(i)$ can be defined as

$$D(i) = \frac{D_{avg}(i)}{n * nc} \qquad (3)$$

The definition of relative normalized delay D(i) effectively

extends the definition of normalized delay in [26] to consider the different priority classes.

***Universal performance factor U(i) of i-class priority traffic,*** where $i$=1..$p$ is defined by a relation involving the two major above normalized factors, $D(i)$ and $Th(i)$*:* the performance of a MIN is considered optimal when $D(i)$ is minimized and $Th(i)$ is maximized, thus the formula for computing the *universal performance factor* arranges so that the overall performance metric follows that rule. Formally, $U(i)$ can be expressed by

$$U(i) = \sqrt{w_d * D(i)^2 + w_{th} * \frac{1}{Th(i)^2}} \qquad (4)$$

where $w_d$ and $w_{th}$ denote the corresponding *weights* for each factor participating in the *U*, 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 *U* becomes smaller, thus smaller *U* 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 (4) can be replaced by:

$$U(i) = \sqrt{w_d * \left(\frac{D(i) - D(i)^{\min}}{D(i)^{\min}}\right)^2 + w_{th} * \left(\frac{RTh(i)^{\max} - RTh(i)}{RTh(i)}\right)^2} \qquad (5)$$

where $D(i)^{min}$ is the minimum value of *normalized packet delay* $D(i)$ and $RTh(i)^{max}$ is the maximum value of *Relative normalized throughput* $RTh(i)$. Consistently to equation (4), when the *universal performance* factor $U(i)$, as computed by equation (5) is close to 0, the performance a MIN is considered optimal whereas, when the value of $U(i)$ increases, its performance deteriorates. Finally, taking into account that the values of both *delay* and *throughput* appearing in equation (5) are normalized, $D(i)^{\min} = RTh(i)^{\max} = 1$, thus the equation can be simplified to:

$$U(i) = \sqrt{w_d * [D(i) - 1]^2 + w_{th} * \left(\frac{1 - RTh(i)}{RTh(i)}\right)^2} \qquad (6)$$

The definition of universal performance U(i) effectively extends the definition of universal performance factor in [19] to consider the different priority classes.

Finally, we list the major parameters affecting the performance of a multi-priority, multi-layered MIN.

- *Number of priority classes p* is the number of different priority classes, where 1 represents the lowest packet class priority, and *p* denotes the highest one. In our study, we consider four distinct priorities, a scheme adopted by a number of commercial switches (e.g. [23], [24], [25]). In [23], the four categories are defined as *low*, *medium*, *high* and *absolute* priority, with absolute priority being mainly used for time-critical control traffic, and the normal data traffic being partitioned into the remaining three categories (e.g. on-line transaction processing: high; backup: low; other traffic: medium). Since time-critical control traffic is low in volume, in this study we merge the *absolute priority* and *high-priority* classes into a single priority class, resulting in a three-class priority scheme with 1-class, 2-class and 3-class standing for low-, medium- and high-priority packets respectively. The merging of the two priority classes allows us to save one additional buffer space that would be devoted to absolute priority packets which would (a) be underutilized, since time-critical control traffic packets are relatively few and (b) increase the cost of the SE, and therefore the cost of the MIN.
- *Buffer-size b(i) of an i-class priority queue*, where $i$=1..$p$ is the maximum number of such packets that the corresponding $i$-class input buffer of a SE can hold. In this paper we consider symmetric-sized double-buffered $b(i)$=2 MINs, where $i$=1..3 and asymmetric-sized implementations with $b(1)$=3, $b(2)$=2 and b(3)=1. It is worth noting that a buffer size of b($i$)=2 is being considered since it has been reported [19] to provide optimal overall network performance: indeed, [19] documents that for smaller *buffer-sizes* $b(i)$=1 *network throughput* drops due to high *blocking probabilities*, whereas for higher *buffer-sizes* $b(i)$=4 or 8 *packet delay* increases significantly (and the SE hardware cost also raises).
- *Offered load λ(i) of i-class priority traffic*, where $i$=1..$p$ is the steady-state fixed probability of such arriving packets at each queue on inputs. It holds that $\lambda = \sum_{i=1}^{p} \lambda(i)$ , where $\lambda$ represents the total arrival probability of all packets. In our simulation $\lambda$ is assumed to be $\lambda$ = 0.1, 0.2… 0.9, 1.
- *Ratio of i-class priority offered load r(i)*, where $i$=1..$p$ expressed by $r(i)=\lambda(i)/\lambda$. It is obvious that $\sum_{i=1}^{p} r(i) = 1$. In this paper we consider (a) a case of a *normal-QoS setup* in which the ratios of high, medium and low priority packets are assumed to be $r(3)$=0.10, $r(2)$=0.30 and $r(1)$=0.60 respectively, and (b) a case of a *high-QoS setup* with the corresponding ratios becoming $r(3)$=0.20, $r(2)$= 0.40 and $r(1)$= 0.40 respectively .
- *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 $n$=10.
- *Number of single-layer stages s* is the number of stages at the single-layer segment of MIN. In this study, we also consider a multi-layer segment at the end of MIN, where the number of layers within each subsequent stage to be doubled, i.e. $nl(i+1) = 2 * nl(i)\ \forall i: s \le i < n$ [$nl(i)$ denotes the number of layers at stage $i$]. Doubling the number of layers in each subsequent stage guarantees that the last segment of the MIN operates in a blocking-free fashion, in the general case however, the number of layers in each stage $i$+1 within the

multi-layer segment is subject to the constraint $nl(i) \leq nl(i+1) \leq 2*nl(i)$ [18]. Under the assumption that the number of layers within each subsequent stage doubles, the n*umber of layers at the final stage l* will be equal to $2^{(n-s)}$. In this work, we consider $s$=8 and therefore $l$=4.

## IV. Simulation and Performance Results

A special-purpose simulator was developed for evaluating the overall network performance of Delta type MINs. This simulator which was developed in C++, and it is capable to operate under different configuration schemes. It supports various input parameters such as the *buffer-length* of high, medium and low priority queues respectively, the *number* of input and output ports, the *number* of stages, the *offered load*, the *ratios* of all priority classes of packets and the *number of layers* of last stage. Internally, each SE of a MIN supporting $p$ priority classes was modeled as an array of $p$ non-shared buffer pairs of queues, with each queue operating in a First-Come-First-Serviced basis and one buffer from each pair dedicated to the upper queuing bank and the other dedicated to the lower queuing bank.

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 a packet from one stage to the next. All packet contentions were resolved by favoring those packets transmitted from the higher priority queues in which they were stored in, while the contention between two packets of the same priority class was resolved randomly.

Metrics such as packet *throughput* and packet *delay* 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.

### A. Simulator Validation

Single-layered single-buffered 6-stage 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 single- and dual-priority schemes. This was done by setting the parameter $p$ (number of priority classes) in our simulator to 1 and 2 respectively. In the case of single-priority traffic $p$=1, we noticed that all simulation experiments were in close agreement with the results reported in [22] (fig. 2 in [22]), and -notably- with Theimer's model [15], which is considered to be the most accurate one. For $p$=2 (dual-priority MINs) we compared our measurements against those obtained from Shabtai's Model reported in [13], and have found that both results are in close agreement (maximum difference was only 3.8%).

### B. Simulation Algorithms

The simulation of the multi-layered, multi-priority MIN effectively involves two processes which run in every SE: the first process scans the queues within the SE to locate a packet that can be forwarded to the next stage; once such a packet is located, the second process is invoked to perform the forwarding. Algorithm 1 displays the details of the queue scanning process, while Algorithm 2 depicts the internals of the second process.

The performance evaluation presented in this paper is independent from the internal link permutations of a banyan-type network (Delta, Omega, Generalized Cube), thus it can be applied to any class of such networks.

*Queue-Process* ($cs_{id}$, $cl_{id}$, $nl_{id}$, $sq_{id}$)
*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
{
processor=0 ;
for ($pr_{id}$=P-1; $pr_{id}$>=0; $pr_{id}$--) // where 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($RA[sq_{id}][cs_{id}][clid][pr_{id}][1]$) ; // get the $(cs_{id})^{th}$ bit of Routing Address (RA)
// for the leading packet of $pr_{id}$-class Send-Queue by a cyclic logical left shift
if ($RA_{bit}$= 0) // upper port forwarding
$aq_{id}$ = 2 * ($sq_{id}$ % (N/2) ) ; // link for perfect shuffle algorithm
// where N is the total number of input/output ports
else // lower port forwarding
$aq_{id}$ = 2 * ($sq_{id}$ % (N/2) ) +1 ; // link for perfect shuffle algorithm
// the above network implementation (omega-type) has the same interconnection links between the crossbar stages
*Unicast-Forwarding* ($cs_{id}$, $cl_{id}$, $nl_{id}$, $sq_{id}$, $aq_{id}$, $pr_{id}$) ;
processor=1 ;
}
}

Algorithm 1: Queue-Process for multi-layered, multi-priority MINs

*Unicast-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; 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

```
{
  if (Pop[aq_id][cs_id+1][nl_id][pr_id] = B[cs_id+1] [pr_id]) // Blocking State;
  // where B[cs_id+1] [pr_id] is the buffer-size of the pr_id-class Accept-Queue of Next Stage cs_id+1
     Blocked[sq_id][cs_id][cl_id][pr_id] = Blocked[sq_id][cs_id][cl_id][pr_id]+1 ;
  else // unicast-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] ; // where RA is the Routing Address
       // of the packet located at (bf_id)^th position of Send-Queue
   }
  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 ;
}
```

Algorithm 2: Unicast-Forwarding for multi-layered, multi-priority MINs

### C. *Multi-layerd Multi-priority MINs with Asymmetric-sized Buffer Queues*

All SL-S-R[*h,m,l*] curves at subsequent diagrams represent the performance of a single-layer 10-stage Delta Network, under a 3-class priority mechanism, when the *buffer-lengths* of all priority-class SEs are $b(i)=2 \ \forall \ i=1..3$, expressing a symmetric double-buffered MIN setup with the ratios of high, medium and low priority packets to be $r(3)=h$, $r(2)=m$ and $r(1)=l$ respectively. Similarly, curves SL-A-R[*h,m,l*] depict the performance of an asymmetric 10-stage Delta Network, where the *buffer-lengths* of high, medium and low priority packets are $b(3)=1$, $b(2)=2$ and $b(1)=3$ respectively.

At this work, we also extend our findings for multi-layered MINs by setting the number of layers at the last stage to be equal to $l=4$, i.e. the first eight 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 8 stages, double-buffered queues are considered, whereas at the last two stages (which are non-blocking), single-buffered single-priority SEs are used, as the absence of blockings removes the need for larger buffers. Consequently, considering in this paper a 10-stage multi-layer MIN, with four layers at the final stage, it consists of 7168 SEs in overall (4 layers * 512 SEs/layer =2048 SEs for the final stage + 2 layers * 512 SEs/layer = 1024 SEs for the 9$^{th}$ stage + 8 stages * 512 SEs/stage =4096 SEs), an increase of 40% as compared with the 5120 SEs needed for the implementation of a single-layer 10-stage MIN (10 stages * 512 SEs/stage = 5120 SEs). Since each SE at last 2 stages of multi-layered segment needs only 2 buffers to be implemented as compared to a SE of single-layer segment needing 6 buffer units the buffer-space increment is confined to 13.3%. Finally, in the following paragraphs, the prefix of ML- at the begging of curve names declares multi-layer MIN configurations with 4 layers at the last stage (as opposed to prefix SL-, which denotes single-layer setups).

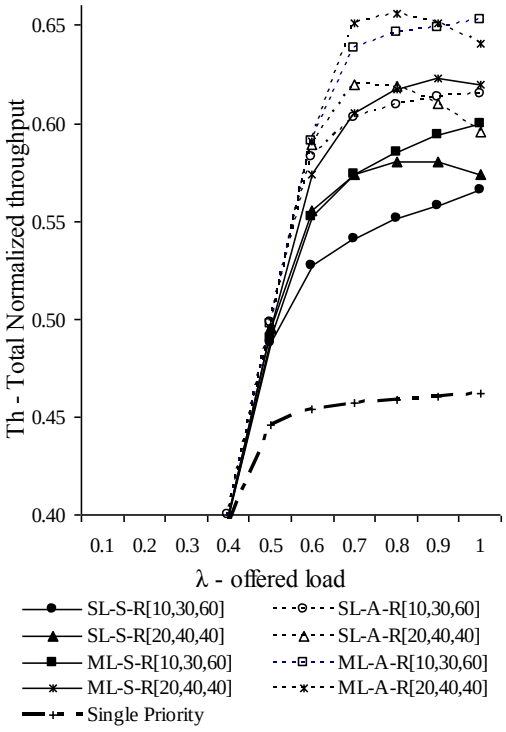


FIGURE 2: Total normalized throughput vs. offered load

Figure 2 depicts the simulator results obtained regarding the total normalized throughput for various MIN configurations. The segment corresponding to offered loads between $\lambda=0.1$ and $\lambda=0.4$ has been omitted from the figure to provide better

detail for the load range between λ=0.4 and λ=1; all curves in the omitted range increase linearly with the offered load since, at this load range, the network has amble switching power to fully service the offered load. According to figure 2 the gains for *total normalized throughput* of a symmetric-sized double-buffered Delta Network, employing a single-layer multi-class priority mechanism (curves SL-S-R[*h*,*m*,*l*]) vs. the corresponding single priority one are 22.5% and 26.4%, under a normal-QoS (*h*=0.10, *m*=0.30, *l*=0.60) and a high-QoS (*h*=0.20, *m*=0.40, *l*=0.40) setup, when $\lambda$=1 and $\lambda$=0.8 respectively. The performance improvement in the overall network throughput may be attributed to the exploitation of the additional buffer spaces available for the MIN, since now each priority class has distinct buffer spaces and thus blockings due to buffer space unavailability occur with decreased probability.

Note that when asymmetric-sized MINs (curves SL-A-R[*h*,*m*,*l*]) are implemented the corresponding gains are further improved, rising to 33.2% and 35.7%, under normal-QoS and high-QoS setups respectively. This can be attributed to improved buffer space exploitation, since in the symmetric-sized case high-priority buffers are under-utilized because (a) high priority packets are ;less in number and (b) high priority packets are immediately forwarded when present, therefore queuing will occur *only* if a contention at the receiving SE appears; for medium- and low-priority packets queuing will occur when *either* a high-priority packet is serviced *or* when contention at the receiving SE appears.

Finally, expanding all previous configurations by introducing multi-layer (*l*=4) schemes the gains of all setups were considerably improved further. For the case of asymmetric-sized MINs (curves ML-A-R[*h*,*m*,*l*]), the improvements were quantified to 41.3% and 42.9% under normal-QoS and high-QoS setups respectively.

Figure 3 depicts the *relative normalized throughput* of high priority packets at single-layer MIN setups. The segment corresponding to offered loads between λ=0.1 and λ=0.5 has been omitted from the figure to provide better detail for the load range between λ=0.5 and λ=1; all curves in the omitted range increase linearly with the offered load since, at this load range, the network has amble switching power to fully service the offered load of high-priority packets. According to this figure all curves approach the optimal throughput value $Th^{max}$=1. Since the *buffer-length* for high priority packets is *b*(*3*)=2 in the case of symmetric-sized MINs (curves SL-S-R[*h*,*m*,*l*]) it is obvious that the *relative normalized throughput* appears to be further improved, but the gains are marginal (7% for the high-QoS setup at full load). Note that the corresponding multi-layer MINs exhibit approximately the same behavior at the case of high priority packets and thus they are not presented at this diagram.

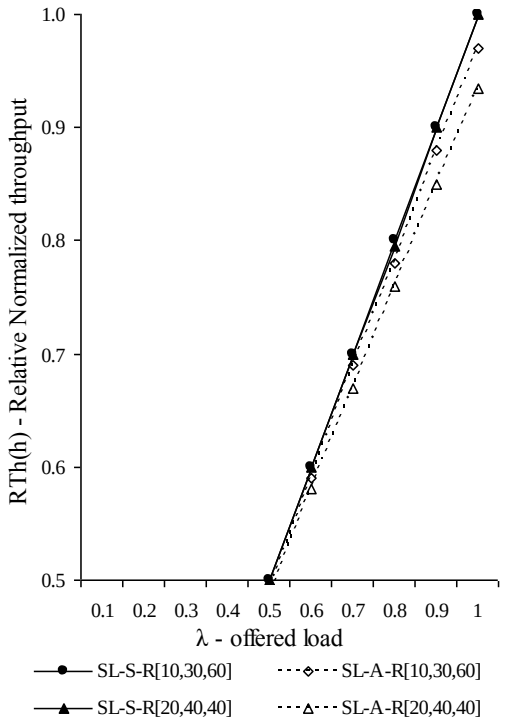


FIGURE 3: Relative normalized throughput of high priority packets vs. offered load

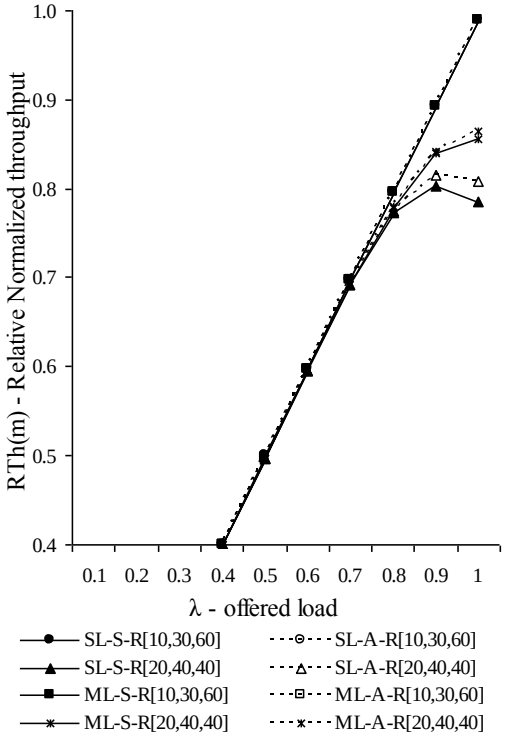


FIGURE 4: Relative normalized throughput of medium priority packets vs. offered load

Figure 4 presents the *throughput* of medium-priority load. The segment corresponding to offered loads between λ=0.1

and λ=0.4 has been omitted from the figure to provide better detail for the load range between λ=0.4 and λ=1; all curves in the omitted range increase linearly with the offered load since, at this load range, the network has amble switching power to fully service the offered load of medium-priority packets. It is obvious that the *relative normalized throughput* of medium priority-class packets is approaching the optimal value $Th^{max}$=1, under all normal-QoS configuration setups. Under these setups, the *buffer-length* for medium priority packets (which is just *b*(*2*)=2 for both symmetric- and asymmetric-sized queue implementations) is adequate to extirpate the effects of collisions of this priority-class packets. On the other hand, at high-QoS setups (*h*=0.20, *m*=0.40, *l*=0.40) the introduction of multiple layers at last two stages (curves ML-S-R[20,40,40] and ML-A-R[20,40,40]) improves the *throughput factor* at higher *offered loads*, where the implementation of asymmetric-sized queues has a small edge over the symmetric-sized one. This marginal improvement can be justified by considering that in the asymmetric configuration, the probability that a higher-priority packet exists at the queue decreases, and hence the probability that a medium-priority packet will be serviced increases.

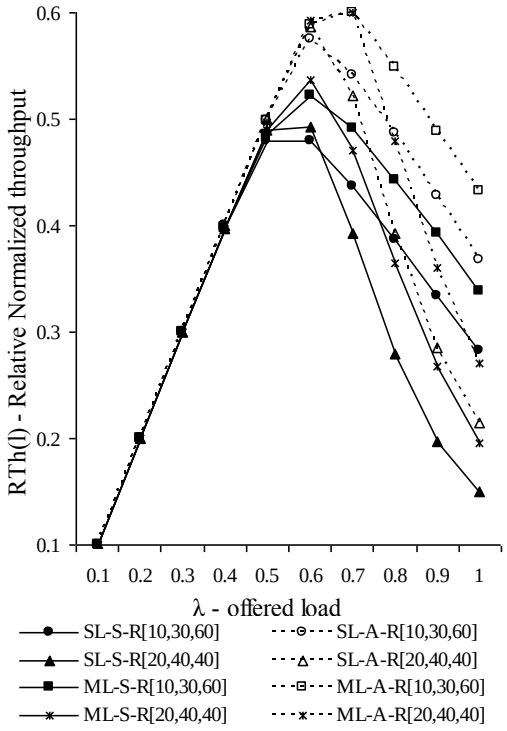


FIGURE 5: Relative normalized throughput of low priority packets vs. offered load

Figure 5 depicts the case of low-priority packet *throughput*. We can observe that the *relative normalized throughput* of low priority packets is considerably better in all asymmetric-sized configurations, where the *buffer-length* for low priority packets is *b*(3)=3, as compared to the symmetric case of having double-buffered queues, for all priority class packets. It is obvious that the asymmetric-sized buffer setup offers superior service to the low-priority packets as compared to the symmetric-sized scheme, mainly owing to the one additional buffer position available in the asymmetric setup to packets of this class. We can also observe that the gains of *throughput* are considerable at moderate and high network loads ($\lambda \geq 0.5$) for all asymmetric-sized setups. Finally, at the case of multi-layer MINs this performance metric exhibits considerable improvement rates, as compared to the corresponding single-layer setups.

Figures 6, 7 and 8 present the findings for the *normalized delay* performance metric for high-, medium- and low-priority packets respectively. In figure 6 we can observe that the performance metric of *normalized delay* for both *equal-sized* buffer and *asymmetric-sized* buffer scheme, where the *buffer-size* for high-priority packets is *b*(3)=2 and *b*(3)=1 respectively, is close to the optimal value $D_{nin}$=1 under both normal- and high-QoS configuration setups. We can also notice that the *asymmetric-sized* scheme has a small edge over the *symmetric-sized* one since the first implementation employs only one buffer unit and consequently shorter queuing delays (at the expense of throughput, cf. fig 3.). For brevity reasons, we do not include a diagram for the multi-layer configuration; most measurements coincide with those illustrated in Figure 6 for the single-layer counterpart configurations, with few exceptions deviating by 0.01 or 0.02.

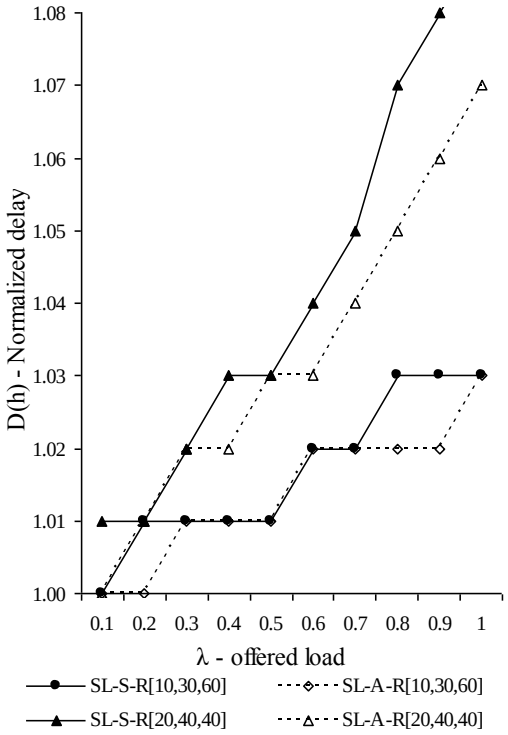


FIGURE 6: Normalized delay of high priority packets vs. offered load

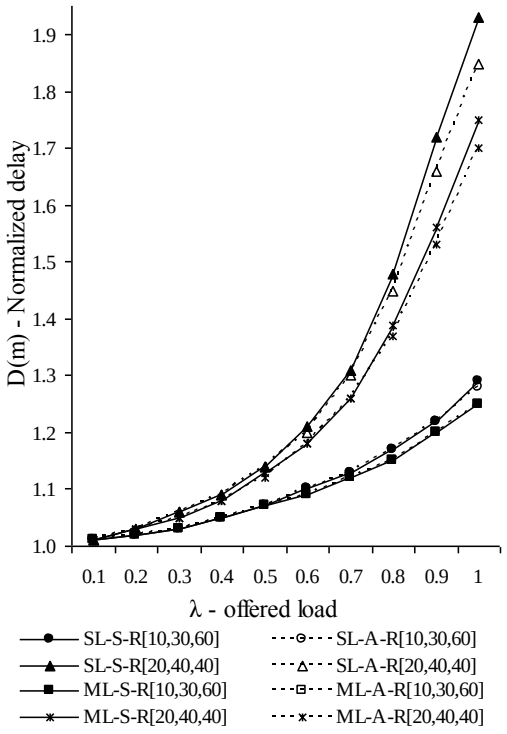


FIGURE 7: Normalized delay of medium priority packets vs. offered load

In figure 7 we can notice that *normalized delay* exhibits approximately identical behaviour for both symmetric- and asymmetric-sized configurations, similarly to the case of normalized throughput for medium-priority packets. We can also observe that using a multi-layer scheme at last two stages, the performance metric of *delay* is slightly improved at both normal- and high-QoS configuration setups due to the fact that there is no blockings at these stages. Finally, when comparing the delay in the *normal-QoS* setup against the delay in the *high-QoS* configuration, we notice that in the case of the *high-QoS* configuration we have a increment in the range of 35% to 50% (at full load) against the corresponding *normal-QoS* configurations. This deterioration is expected due to (a) the presence of more high-priority packets in the network and (b) the increased contention between normal-priority packets, which are now greater in number.

Figure 8 depicts the *normalized delay* for low-priority packets. Providing one additional buffer unit to low-priority packets at asymmetric-sized scheme in order to have a better *throughput* performance, it is observed that *normalized delay* factor deteriorates by 18.8% and 12.1% (under full load traffic conditions) when normal-and high-QoS setup of single-layered MINs is employed respectively. On the other hand, in the case of normal-QoS setup the *normalized delay* metric is improved 8.1% and 6.8% by applying a multi-layer scheme at the last two stages of MIN, when an *equal-sized* and *asymmetric-sized* buffer scheme is employed respectively. It is also worth noting that the gain of *normalized delay* for the second scenario of a high-QoS setup is similar to previous one, but it is maximized when the offered load of multi-layered MINs is (λ=0.9).

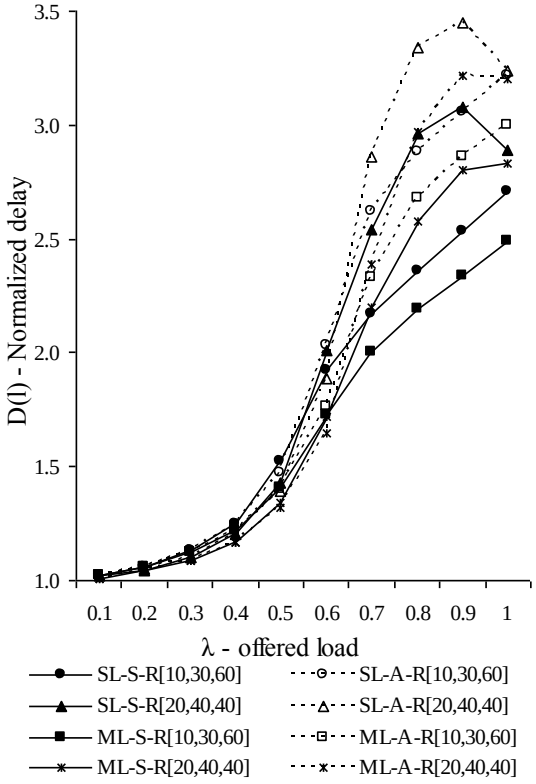


FIGURE 8: Normalized delay of low priority packets vs. offered load

Figures 9, 10 and 11 depict the universal performance factor for the different setups, for high-, medium- and low-priority packets respectively. The segment corresponding to low offered loads (λ=0.1 to λ=0.2) has been omitted from these figure to provide better detail for the load range between λ=0.2and λ=1; for the load range λ=0.1 to λ=0.2 the universal performance factor exhibits very high values, since the network is underutilized regarding its relative throughput and therefore the second term $\left(\frac{1-RTh(i)}{RTh(i)}\right)$ dominates the universal performance factor equation (cf. [25]).

Regarding the high-performance packets, we can notice that the universal performance factor is very close for all setups and actually improves (acquires smaller values) as the offered load increases, because the network bandwidth is better exploited at higher loads, leading thus to higher normalized throughput values.

In figure 10 we can notice that the universal performance factor for medium-priority packets improves up to the load of 0.6-0.8 (depending on the setup examined), and subsequently deteriorates. This is due to the fact that at the first segment of offered load (0.1-0.7) the improvement in normalized throughput has a higher impact to the universal performance factor than the respective deterioration in the delay; at higher

loads, however, normalized throughput improves less (or even deteriorates), while the delay continues to rise.

The same remarks hold for the case of low-priority packets (figure 11), at this case however the optimal value of universal performance factor is attained at a smaller load (0.5-0.6).

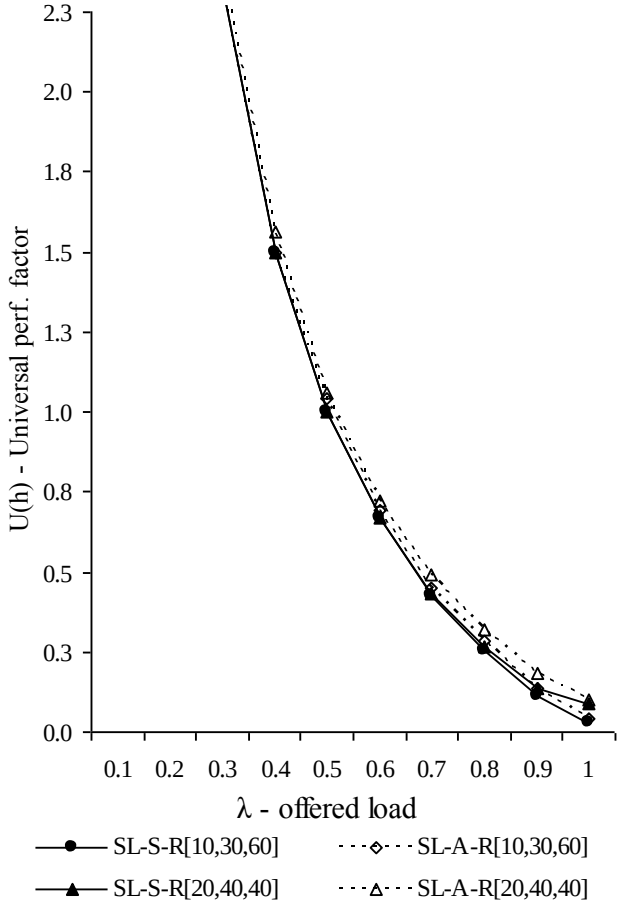


FIGURE 9: Universal performance factor of high priority packets vs. offered load

Regarding the difference between the symmetric vs. asymmetric buffer sizing, we can observe that the asymmetric setup has a considerable performance edge over the symmetric one. For normal-priority packets this only becomes apparent in the high-QoS setup and for high offered loads ($\lambda \geq 0.8$), but for low-priority packets the performance edge of asymmetric buffer sizing is obvious for both normal- and high-QoS configurations and for loads $\lambda \geq 0.6$.

Finally, regarding the introduction of multiple layers at the final stages of the MIN, expectedly the multi-layer MINs exhibit higher performance than their single-layer counterparts; however, these gains are only considerable in the case of the high-QoS setup and particularly for the low-priority packets. Therefore, considering the increased cost of multi-layer configurations, it might not be worthwhile to employ multiple layers unless the throughput of low-priority packets is a major concern.

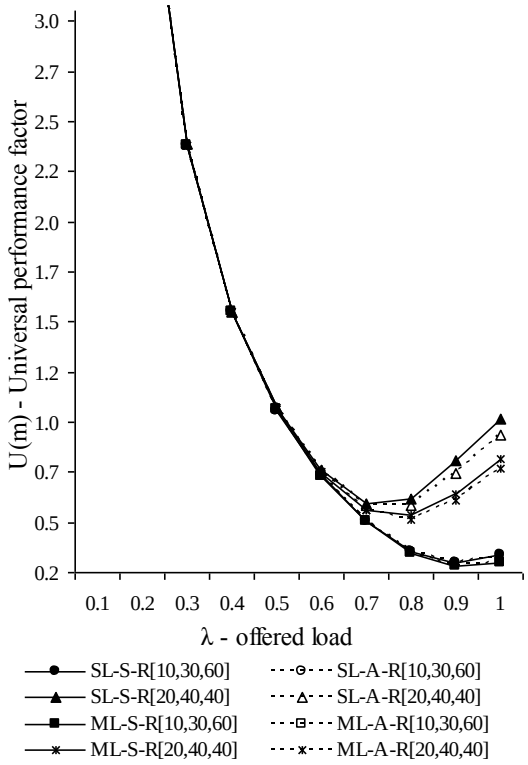


FIGURE 10: Universal performance factor of medium priority packets vs. offered load

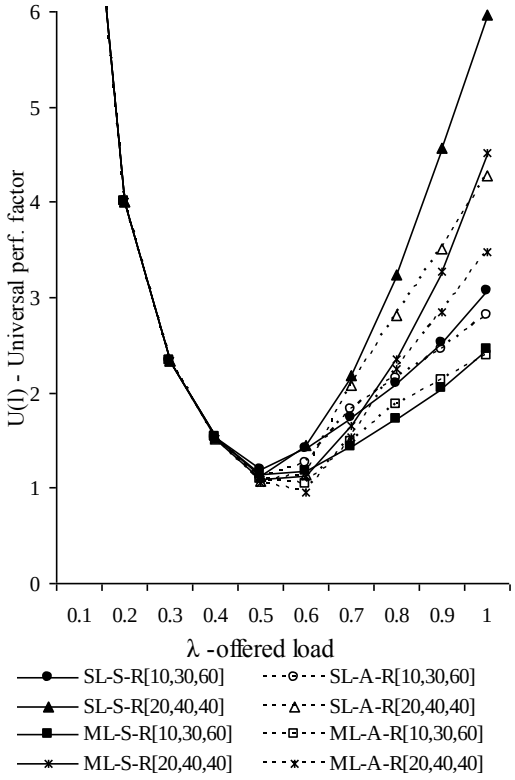


FIGURE 11: Universal performance factor of low priority packets vs. offered load

## V. CONCLUSIONS

In this paper we have studied and compared the performance of an *asymmetric buffer size* configuration for multi-class priority MINs combined with the introduction of a multi-layered segment at last stages against the typical single-layered *equal-sized buffer* MIN configuration under different traffic loads.

The *asymmetric-sized buffer* configuration has been found to better exploit network resources and capacity, since the available buffers can be more appropriately allocated to the priority class that needs them. More specifically, we found that the *asymmetric buffer size* configuration provides better overall *throughput* against its *equal-sized buffer* counterpart. The *asymmetric-sized* buffer configuration achieves these performance benefits because it better matches buffer allocation to the shape of network traffic. Examining the three different priority classes of *offered load* in more detail, we noticed that the *asymmetric buffer size* scheme provides significantly better *throughput* and *delay* for low-priority packet and slightly better performance for medium-priority packets when the load of input packets is high. On the other hand, for high-priority packets the performance of the two schemes is almost identical, with the *equal-sized buffer* scheme having a small edge.

In this work we have also extended the *asymmetric buffer size scheme* as a solution to the problem of performance degradation of lower priority packets by introducing a multi-layer architecture and improving furthermore their performance. Since multi-layer architectures are associated with higher costs, we have limited the multi-layer portion of the network to the final two stages (over a total of ten stages), balancing thus between performance and cost. It is worth noting that performance gains were found again to be considerable; both in terms of *throughput* and *delay*. Moreover, the multi-layered implementation can also support trunked multicasting at last non-blocking stages without any degradation.

Consequently, 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.

As part of our future work, we consider the examination of different arrival processes, including bursty arrivals, Markov-modulated poisson processes and fluid traffic models [27]. Performance evaluation under multicast and hotspot traffic patterns will be also considered.

## REFERENCES


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

[2] C. Bauer, "Throughput and Delay Bounds for Input Buffered Switches Using Maximal Weight Matching Algorithms and a Speedup of Less than Two*", Information Networking*, LNCS , vol. 3090, pp. 658-668, 2004

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

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

[5] J. Garofalakis, E. Stergiou, "An approximate analytical performance model for multistage interconnection networks with backpressure blocking mechanism", Journal of Communications (JCM), Academy, vol. 5, no 3, pp. 247-26, March 2010

[6] J. Garofalakis, and E. Stergiou "An analytical performance model for multistage interconnection networks with blocking", Proceedings of Communication Networks and Services Research Conference CNSR'08, IEEE Press, pp. 373-380, May 2008.

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

[8] D. A. Lawrie. "Access and alignment of data in an array processor", *IEEE Transactions on Computers*, vol. 24, no. 12, pp.1145-1155, Dec. 1975.

[9] 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, pp. 68 - 78, 1991.

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

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

[12] M. Roughan, S. Sen, O. Spatscheck, N. Duffield, "Class-of-Service Mapping for QoS: A Statistical Signature-based Approach to IP Traffic Classification", *Procs. of IMC'04*, October 25–27, Taormina, Sicily, Italy, pp. 135-148, 2004

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

[14] Stevens W. R., "TCP/IP Illustrated", vol 1. *The protocols, (10th Ed), Addison-Wesley Pub Company*, 1997.

[15] 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, February 1991.

[16] 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.

[17] D.Tutsch, G.Hommel. "Generating Systems of Equations for Performance Evaluation of Buffered Multistage Interconnection Networks", *Journal of Parallel and Distributed Computing*, 62, no. 2, pp. 228-240, 2002.

[18] 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.

[19] D.C. Vasiliadis, G.E. Rizos, and C. Vassilakis. "Performance Analysis of blocking Banyan Swithces", *Proceedings of the IEEE-sponsored International Joint Conference on Telecommunications and Networking CISSE 06*, December, pp. 107-111, 2006.

[20] D.C. Vasiliadis, G.E. Rizos, C. Vassilakis, and E.Glavas. "Performance evaluation of two-priority network schema for single-buffered Delta Network", *Proceedings of the 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07)*, art. no. 4394153, 2007

[21] D.C. Vasiliadis, G.E. Rizos, C. Vassilakis. "Improving Performance of Finite-buffered Blocking Delta Networks with 2-class Priority Routing through Asymmetric-sized Buffer Queues", Proceedings of the Fourth

Advanced International Conference on Telecommunications AICT08, IEEE Press, pp.23-29, 2008

[22] D.C. Vasiliadis, G.E. Rizos, C. Vassilakis. "Performance Study of Multi-Layered Multistage Interconnection Networks under Hotspot Traffic Conditions", *Journal of Computer Systems, Networks, and Communications, Hindawi Publishing Corporation*, Vol. 2010, art.id 403056, 11 pages

[23] Cisco Systems. Cisco MDS 9000 Family Fabric Manager Configuration Guide, Release 1.3 (chapter 27). Available at http://www.cisco.com/en/US/docs/storage/san_switches/mds9000/sw/rel_1_x/1_3/fm/configuration/guide/fm_cg.html

[24] Hewlet-Packard. HP ProLiant BL e-Class C-GbE Interconnect Switch User Guide. Available at http://h20000.www2.hp.com/bc/docs/support/SupportManual/c00594291/c00594291.pdf

[25] Linksys. Linksys SRW224P 24-port 10/100 + 2-port Gigabit Switch - WebView/PoE. Available at http://www.cisco.com/en/US/products/ps9988/index.html

[26] Y. C. Jenq. Performance Analysis of a Packet Switch Based on Single-Buffered Banyan Network. IEEE Journal On Selected Areas In Communications, VOL. SAC-1, NO. 6, December 1983, pp. 1014-1021.

[27] V. S. Frost and B. Melamed. Traffic Modeling For Telecommunications Networks. IEEE Communications Magazine, March 1994, pp. 70-81.