Routing and Performance Analysis of Double-Buffered Omega Networks Supporting Multi-Class Priority Traffic

D. C. Vasiliadis,[a,b] G. E. Rizos,[a,b] C. Vassilakis,[a] E. Glavas[b]
[a]*Department of Computer Science and Technology, University of Peloponnese, Greece*
[b]*Technological Educational Institute of Epirus, Greece*
dvas@uop.gr, georizos@uop.gr, costas@uop.gr, eglavas@teiep.gr

Abstract

In this paper the modeling of Omega Networks supporting multi-class routing traffic is presented and their performance is analyzed. We compare the performance of multi-class priority mechanism against the single priority one, by gathering metrics for the two most important network performance factors, namely packet throughput and delay under uniform traffic conditions and various offered loads, using simulations. Moreover, two different test-bed setups were used in order to investigate and analyze the performance of all priority-class traffic, under different Quality of Service (QoS) configurations. In the considered environment, Switching Elements (SEs) that natively support multi-class priority routing traffic are used for constructing the MIN, while we also consider double-buffered SEs, two configuration parameters that have not been addressed insofar. The rationale behind introducing a multiple-priority scheme is to provide different QoS guarantees to traffic from different applications, which is a highly desired feature for many IP network operators, and particularly for enterprise networks.

1. Introduction

Multistage Interconnection Networks (MINs) with crossbar Switching Elements (SEs) are frequently employed in multiprocessor computer architectures for interconnecting processors and memory modules [5, 20]. MINs are also increasingly used for implementing the switching fabric of high-capacity communication processors, such as ATM switches, gigabit Ethernet switches and terabit routers [2, 3, 4].

The popularity of MINs stems from both the operational features they deliver –e.g. their ability to route multiple communication tasks concurrently- and the appealing cost/performance ratio they achieve. MINs with the Banyan [1] property e.g. Omega Networks [7], Delta Networks [6], and Generalized Cube Networks [8] are more widely adopted, since non-Banyan MINs have -generally- higher cost and complexity.

Both in the context of parallel and distributed system, the performance of the communication network interconnecting the system elements (nodes, processors, memory modules etc) is recognized as a critical factor for overall system performance. Consequently, the need for communication infrastructure performance prediction and evaluation has arisen, and numerous research efforts have targeted this area, employing either analytical models (mainly based on Markov models and Petri-nets) or simulation techniques.

The past few years have witnessed a dramatic increase in the number and variety of applications running over the Internet and over enterprise IP networks. The spectrum includes interactive (e.g. telnet, and instant messaging), bulk data transfer (e.g. ftp, and P2P file downloads), corporate (e.g. database transactions), and real-time applications (e.g. voice, and video streaming). These application classes have considerably different requirements from the communication infrastructure in terms of quality of service aspects, such as *throughput*, *delay* or jitter (e.g. bulk transfer applications need high throughput, interactive applications need minimal delays while streaming applications require bounded jitter), and these requirements are typically expressed to the network layer in the form of packet priorities. Another source of packet priority differentiation is protocol intrinsics, such as TCP out-of-band/expedited data, which are normally prioritized against normal connection data [9]. In order to address these requirements, dual priority (or 2-class) queuing systems have been recently introduced in MINs, providing the ability to offer different QoS parameters to packets that have different priorities.

Several commercial switches already have accommodated traffic priority schemes, such as [15, 16]. Each switching element in these products' fabric has two queues at each input port, with one queue dedicated to high priority packets and the other dedicated to low priority ones. High priority packets are serviced first, while low-priority packets are only serviced when no high-priority packets contend for the same resources (output links). While it is obvious that high-priority packets will receive better quality of service than low-priority ones, the performance of dual priority MINs has not been adequately investigated insofar in order to quantify gains and losses under various traffic conditions, and only few results (e.g. [12, 14]) have been published.

The MINs used in the above studies employ single-buffered SEs, where one buffer position is dedicated to low priority packets and one buffer position is assigned to high priority traffic. In corporate environments, however, hosting a multitude of applications, a two-priority scheme is bound not to suffice for expressing the diversity of application-level requirements to the network layer. As identified in [17], 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., Oracle 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.

In this paper we extend the previous studies by introducing MINs that natively support multi-class routing traffic. We also analyze the performance of multi-priority SEs that use not only single-buffered, but also double-buffered queues in order to offer better QoS, while providing in parallel better overall network performance.

The remainder of this paper is organized as follows: in section 2 we briefly analyze an Omega 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.

2. Omega Network Description for Multi-Class Priority Traffic

There are three major classes of blocking Multistage Interconnection Networks (MINs): Delta Networks which were proposed by Patel [6], Omega Networks [7] and Generalized Cube Networks [8]. All these multistage interconnection self-routing networks are characterized by the fact that there is exactly a unique path from each input port to each output port, which is just the Banyan property as defined in [1]. Internally, they are constructed by small size Switching Elements (SEs) followed or preceded by links. Switching in these networks is termed as “self-routing” because when a SE accepts a packet in one of its input ports, it can decide to which of its output ports it must be forwarded, depending only on the packet’s destination address. A typical configuration of an N X N Omega Network is depicted in figure 1 and outlined below.

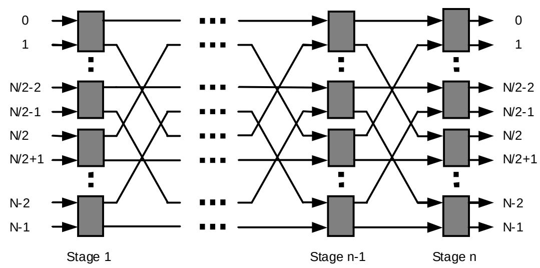

Figure 1. (*N*x*N*) Omega Network

Omega Networks use the “perfect shuffle” routing algorithm by rotating to left only the destination tag. A variation of this algorithm is used by Delta Networks, where a SE of stage k can decide in which output port to send it based on the k^{th} bit of the destination address and the “k-bit shuffle” algorithm, while on Generalized Cube Networks the routing tag is generated by exclusive or of source and destination labels.

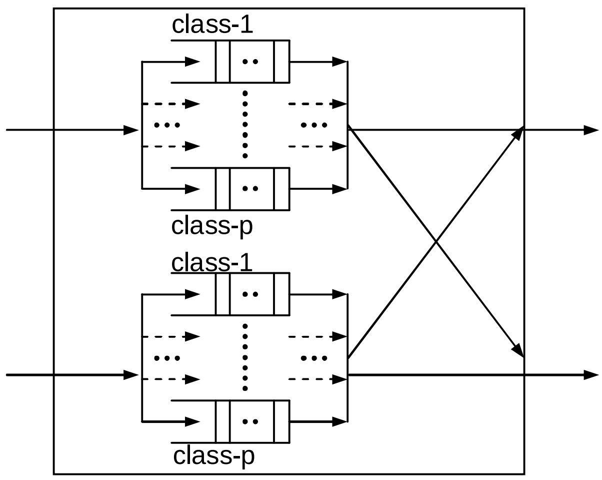

Figure 2. Multi-Priority 2X2 Switching Element

Figure 2 illustrates the internal modeling of a multi-priority SE supporting p priority classes. Each SE is modeled by as 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. 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.

The performance evaluation presented in this paper is independent from the internal link permutations of a banyan-type network, thus it can be applied to any class of such networks. In our study we used an Omega Network that is assumed to operate under the following conditions:

- The MIN operates in a slotted time model [18]. 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.

3. Performance Evaluation Methodology

In order to evaluate the overall performance of a multi-priority (*N*x*N*) MIN consisting of (2x2) SEs, we use the following metrics. Let *T* be a relatively large time period divided into *u* discrete time intervals ($\tau_1, \tau_2, \ldots, \tau_u$).

Average throughput Th_{avg} is the average number of packets accepted by all destinations per network cycle. Formally, Th_{avg} (or *bandwidth*) is defined as

$$Th_{avg} = \lim_{u \to \infty} \frac{\sum_{k=1}^{u} n(k)}{u} \qquad (1)$$

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

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

$$Th = \frac{Th_{avg}}{N} \qquad (2)$$

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 λ(i)* of such packets.

$$RTh(i) = \frac{Th(i)}{l(i)} \qquad (3)$$

Average packet delay $D_{avg}(i)$ of i-class priority traffic, where *i*=1..*p* is the average time a corresponding-class priority packet spends to pass through the network. Formally, $D_{avg}(i)$ is expressed by

$$D_{avg}(i) = \lim_{u \to \infty} \frac{\sum_{k=1}^{n(u)} t_d(k)}{n(u)} \qquad (4)$$

where $n(u)$ denotes the total number of the corresponding-class priority packets accepted within *u* time intervals and $t_d(k)$ represents the total delay for the k^{th} such packet. We consider $t_d(k) = t_w(k) + t_{tr}(k)$ where $t_w(k)$ denotes the total queuing delay for k^{th} packet waiting at each stage for the availability of a corresponding-class empty buffer at the next stage queue of the network.

The second term $t_{tr}(k)$ denotes the total transmission delay for k^{th} such packet at each stage of the network, that is just $n*nc$, where $n=\log_2 N$ is the number of intermediate stages and nc is the network cycle.

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). Formally, $D(i)$ can be defined as

$$D(i) = \frac{D_{avg}(i)}{n * nc} \quad (5)$$

Universal performance 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 factor* arranges so that the overall performance metric follows that rule. Formally, $U(i)$ can be expressed by

$$U(i) = \sqrt{D(i)^2 + \frac{1}{Th(i)^2}} \quad (6)$$

It is obvious that, when the *packet delay* factor becomes smaller or/and *throughput* factor becomes larger the *universal performance* factor $U(i)$ becomes smaller. Consequently, as the *universal performance* factor $U(i)$ becomes smaller, the performance of a MIN is considered to improve. Because the above factors (parameters) have different measurement units and scaling, we normalize them to obtain a reference value domain. Normalization is performed by dividing the value of each factor by the (algebraic) minimum or maximum value that this factor may attain. Thus, equation (6) can be replaced by:

$$U(i) = \sqrt{\left(\frac{D(i) - D(i)^{\min}}{D(i)^{\min}}\right)^2 + \left(\frac{Th(i)^{\max} - Th(i)}{Th(i)}\right)^2} \quad (7)$$

where $D(i)^{min}$ is the minimum value of *normalized packet delay D(i)* and $Th(i)^{max}$ is the maximum value of *normalized throughput Th(i)*. Consistently to equation (6), when the *universal performance* factor $U(i)$, as computed by equation (7) 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 (7) are normalized, $D(i)^{\min} = Th(i)^{\max} = 1$, thus the equation can be simplified to:

$$U(i) = \sqrt{[D(i) - 1]^2 + \left(\frac{1 - Th(i)}{Th(i)}\right)^2} \quad (8)$$

The universal performance factor effectively combines the two most important performance factors into a single metric, which may be used by network designers to directly compare the performance of two network setups. Moreover, network designers may specify *weights* for each factor participating in the universal performance factor, designating thus its importance for the corporate environment; in this way, the performance of a particular MIN setup is expressed in a single metric that is tailored to the needs that the MIN will serve.

Finally, we list the major parameters affecting the performance of a multi-class priority 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 the number of priority classes is assumed to be $p=3$, where 1-class stands for low priority packets, while 2-class and 3-class stand for medium and high priority packets respectively.
- *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 single- $b(i)=1$ or double- $b(i)=2$ buffered MINs. It is worth noting that a buffer size of b=2 is being considered since it has been reported [13] to provide optimal overall network performance: indeed, [13] 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 $l = \sum_{i=1}^{p} l(i)$, where λ represents the total arrival probability of all packets. In our simulation λ is assumed to be $\lambda = 0.1, 0.2\ldots 0.9, 1$.
- *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 the case of a normal-QoS setup 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, while in the case of a high-QoS setup the corresponding ratios become $r(3)=0.20$, $r(2)=0.40$ 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 $n=10$.

4. Simulation and Performance Results

A multi-priority simulator was constructed for evaluating the overall network performance of Omega type MINs. This method of modeling [10] using simulation experiments was applied due to the complexity of the mathematical model [11]. For this purpose we developed a special multi-priority simulator in C++, capable to operate under different configuration schemes. It was based on several input parameters such as the *number of priority classes,* the *buffer-lengths* of queues for all priority classes, the *number* of input and output ports, the *number* of stages, the *offered load*, and the *ratios* of all priority classes of packets. Internally, each SE of a MIN supporting p priority classes was modeled by as an array of p non-shared buffer pairs of queues, with each queue operating in a FCFS 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 *delays* were collected at the output ports. We performed extensive simulations to validate our results. All statistics obtained from simulation running for 10^5 clock cycles. The number of simulation runs was adjusted to ensure a steady-state operating condition for the MIN. There was a stabilization process in order the network be allowed to reach a steady state by discarding the first 10^3 network cycles, before collecting the statistics.

4.1. Simulator validation

Since no other simulator supporting more than two priorities has been reported insofar in the literature, we validated our simulator against single-priority and dual-priority simulators that have been made available. This was done by setting the parameter p (number of priority classes) in our simulator to 1 and 2, and comparing the results obtained from the simulation against results already published for single- and dual-priority MINs. For single-priority MINs, our results have to be found in close agreement with those produced by Theimer's model, which are considered to be the most accurate ones [19].

For p=2 (dual-priority MINs) we compared our measurements against those obtained from Shabtai's Model reported in [12], and have found that both results are in close agreement (maximum difference was only 3.8%). Figure 3 illustrates such a comparison, involving the *total normalized throughput* for both high and low priority packets of a 2-class priority, single-buffered, 6-stage MIN vs. the *ratio of high priority packets* under full *offered load* conditions (p=2 and $b(i)$=1, where i=1..p).

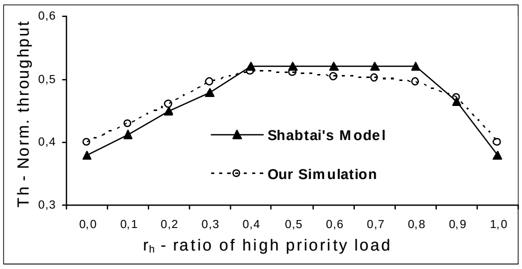

Figure 3. Total normalized throughput of a dual-priority, single-buffered, 6-stage MIN

4.3. Multi-priority MINs

In this paper we extend our study by introducing multi-priority SEs that use not only single-buffered, but also double-buffered queues in order to offer better quality-of-services, while providing in parallel better overall network performance.

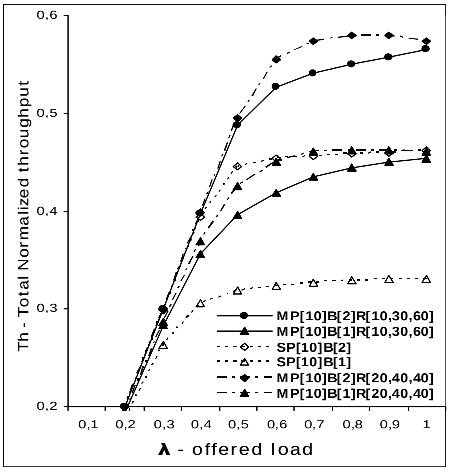

Figure 4. Total normalized throughput of multi-priority MINs vs. offered load

In figure 4, curves MP[10]B[b]R[h,m,l] represent the *total normalized throughput* of a 10-stage Omega Network, under a multi-priority mechanism, when the *buffer-lengths* of all priority-class SEs are $b(i)$=1 or 2, where i=1..p, expressing a symmetric single- or 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 SP[10]B[b] depict the *normalized throughput* of a 10-stage Omega Network, under a single priority mechanism, when the *buffer-length* is b=1 or 2.

According to this figure the gains for *total normalized throughput* of a single-buffered Omega Network, employing a multi-class priority mechanism (curves

MP[10]B[1]R[*h*,*m*,*l*]) vs. the corresponding single priority one (curve SP[10]B[1]) are 37.6% and 41%, 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 λ=1 and λ=0.7 respectively. Similarly, the gains for *total normalized throughput* of a double-buffered Omega Network, employing a multi-class priority mechanism (curves MP[10]B[2]R[*h*,*m*,*l*]) vs. the corresponding single priority one (curve SP[10]B[2]) are 22.5% and 26.4%, under a normal-QoS and a high-QoS setup, when λ=1 and λ=0.8 respectively. The performance improvement in the overall network *throughput* may be attributed to the exploitation of the additional buffer spaces in the MIN, since now each priority class has distinct buffer spaces and thus blockings due to buffer space unavailability occur with decreased probability.

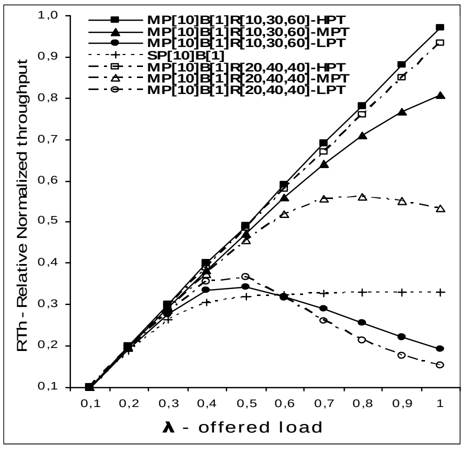

Figure 5. Relative norm. throughput of multi-priority, single-buffered MINs vs. offered load

Figures 5 and 6 depict the *relative normalized throughput* of all priority-class traffic, where –HPT, -MTP, and –LPT stand for high, medium and low priority traffic respectively. According to these figures the *relative normalized throughput* of high priority packets approaches the optimal value of this performance metric at both cases of single-buffered MINs (*Th*=0.97 and 0.93 for a normal- and high- QoS setup respectively), while at both cases of double-buffered configurations it has found to be more improved reaching the maximum value (Th_{max}=1). Medium-priority packets also achieve higher *throughput*, as compared to packets in a single-priority MIN, and it is worth noting that in a normal-QoS double-buffered MIN, this *throughput* approaches the optimal value. Low-priority packets, finally, are receiving *better throughput* than packets in a single-priority MIN when the offered load is less than 0.6, while this service deteriorates when the load exceeds this value. This happens because under heavy loads the probability that packets with high or medium priorities are available at an SE increases, and these packets are chosen over low-priority ones for forwarding to the next MIN stage.

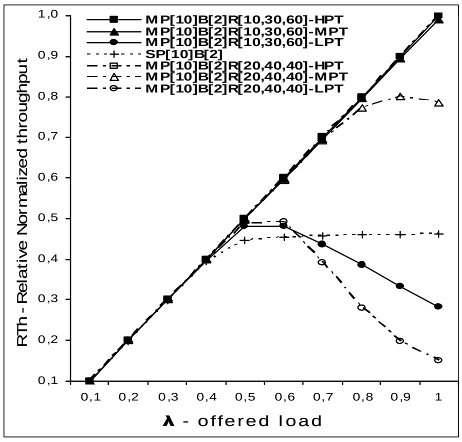

Figure 6. Relative norm. throughput of multi-priority, double-buffered MINs vs. offered load

It worth mentioning that, although the *relative normalized throughput* for all classes of traffic are better for the case of the normal-QoS setup (figures 5 and 6), the *total normalized throughput* is greater at the case of the high-QoS configuration (figure 4), because there are more high and medium priority packets at input ports and thus available buffers are better exploited. In these figures we may see that the MIN can guarantee an QoS regarding throughput for high priority packets in all examined configurations and under all loads. Medium priority packets experience a deterioration in the offered throughput in medium and high loads when a single buffer is available, the addition of extra buffer place however improves the specific metric considerably.

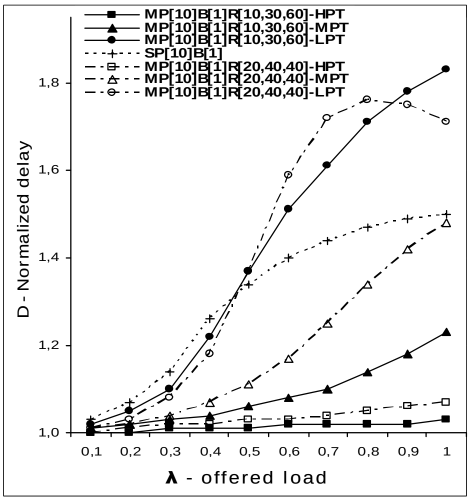

Figure 7. Normalized delay of multi-priority, single-buffered MINs vs. offered load

Figures 7 and 8 represent the findings for the *normalized packet delay* of single- and double-buffered MINs, supporting multi-priority traffic. Again we can observe that high-priority packets obtain service close to the optimal one, especially for the case of a normal-QoS MIN. The *delay* for medium-priority packets is consistently smaller than the *delay* of packets in single-priority networks with equal load; the obtained benefit for this packet class is higher in normal-QoS setups than in high-QoS ones, and this is expected since in the high-QoS setup (a) a considerable amount of network resources is consumed by high-priority packets and (b) more medium-priority packets contend for the remaining network resources.

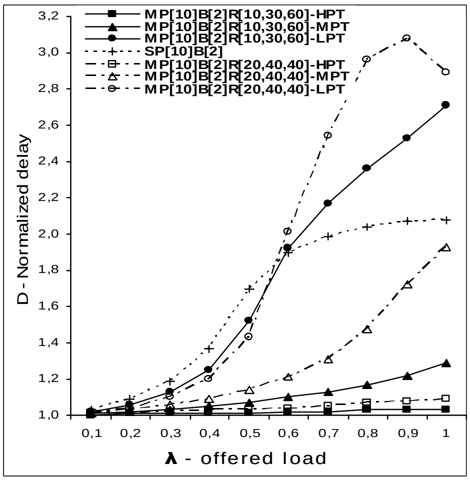

Figure 8. Normalized delay of multi-priority, double-buffered MINs vs. offered load

The *delay* for LPT packets is smaller than *packet-delay* in single-priority MINs for load λ<0.6 in the single-buffer case and λ<0.5 for the double-buffered setup, but subsequently rises since less network slots are available for serving LPT packets (due to higher probabilities that a high- or medium-priority packet exists at an SE). It is worth noting that for this load range, *all* packet classes have smaller delays than the packets in single priority MINs. This may seem contradictory with other works e.g. [13] which report that increments in buffer sizes leads to increased delays; we consider however that in a multi-priority MIN, packets with different priorities are stored in separate queues in SEs, decreasing thus the number of blockings of low- and medium- priority packets due to unavailability of suitable buffer space in the destination SE. At the load range in question, the gains obtained due to the avoidance of these blockings are higher than the costs incurred from yielding to high priority packets, thus the overall effect on the delay from the introduction of distinct buffers for each priority class is positive.

In the curves corresponding to high-QoS setups we can observe a drop in the *delay* for very high loads (λ>0.8 for single-buffered MINs and λ>0.9 for double-buffered MINs): this is due to a high amount of blockings for LPT packets at the input of the MIN's first stage, which effectively preclude a considerable amount of LPT packets from entering the MIN altogether. These packets are not accounted for in the computation of the *delay* metric, and this is the reason why this "improvement" in the performance indicator appears.

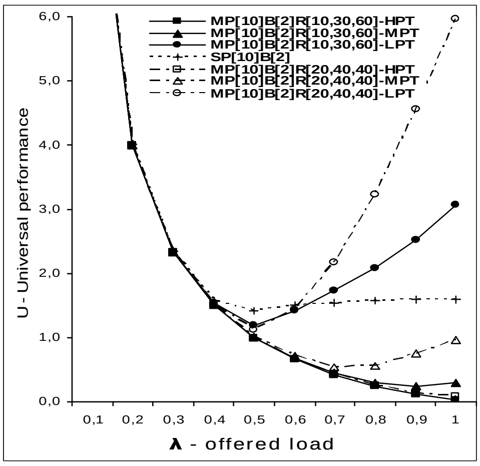

Figure 9. Universal performance of multi-priority, single-buffered MINs vs. offered load

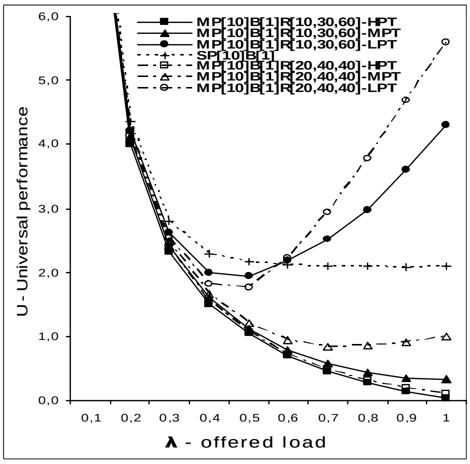

Figure 10. Universal performance of multi-priority, double-buffered MINs vs. offered load

Figures 9 and 10 depict the behavior of the *universal performance factor* metric for each priority-class packets of single- and double-buffered MINs, respectively, in correlation to the *offered load*. The behavior of the *universal performance factor* is follows the behavior of the individual performance indicators, showing that high- and medium-priority packets are offered consistently a better quality of service, as compared to packets in single-priority MINs, while for low-priority packets two areas may be identified: the first one spans along the "light load" segment of the x-axis, in which low-priority packets are offered a better quality of service than packets in single-priority MINs, and the second one spans along the medium- and high-load segment of the x-axis, in which the QoS offered to low-priority packets is inferior to the QoS offered in packets within single-priority MINs. As explained above, the ability to offer better quality of service to all packets at a certain load range is attributed to the existence of more buffers (which are specialized for each priority class); extra buffer availability leads in turn to less blockings, and thus increased *throughput* and smaller *delays*.

5. Conclusions

In this paper we have presented the modeling of multi-priority MINs and analyzed the performance of a MIN supporting three priority classes under various load conditions and two different ratios of QoS requirements.

The goal of this paper is to provide network designers with insight on how packet prioritization affects the QoS delivered to each priority class and the network performance in general. This insight can help network designers to assign packet priorities to various applications in a manner that will comply with the corporate policy, satisfy application requirements and maximize network utilization. The presented results also facilitate performance prediction for multi-priority networks 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 different high/low priority ratios including time-varying loads, both in terms of overall load and priority proportions. Use of non-uniform buffer sizes for different priority classes will be studied as well.

6. References

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

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

[3] Toshio Soumiya, Koji Nakamichi, Satoshi Kakuma, Takashi Hatano, and Akira Hakata. "The large capacity ATM backbone switch FETEX-150 ESP". *Computer Networks*, 31(6), pp. 603–615, 1999.

[4] Ra'ed Y. Awdeh and H. T. Mouftah. "Survey of ATM switch architectures". *Computer Networks and ISDN Systems* 27, pp. 1567–1613, 1995.

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

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

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

[8] 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

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

[10] D. Tutsch, M.Brenner. "MIN Simulate. A Multistage Interconnection Network Simulator" *Procs*. of *17th European Simulation Multiconference (ESM'03);* Nottingham, SCS, pp. 211-216, 2003.

[11] 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.

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

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

[14] 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

[15] D-Link. *DES-3250TG 10/100Mbps managed switch*. ftp://ftp.dlink.eu/datasheets/DES-3250TG.pdf

[16] Intel Corporation. *Intel Express 460T standalone switch*. http://www.intel.com/support/express/switches/460/

[17] 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, 2004, Taormina, Sicily, Italy.

[18] 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 Volume 3090/2004, pp. 658-668.

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

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