Improving performance of finite-buffered blocking Delta Networks with 2-class priority routing through asymmetric-sized buffer queues

D. C. Vasiliadis , G. E. Rizos , C. Vassilakis
Department of Computer Science and Technology
Faculty of Sciences and Technology
University of Peloponnese
GR-221 00 Tripolis Greece

Abstract— **In this paper the performance of asymmetric-sized finite-buffered Delta Networks with 2-class routing traffic is presented and analyzed in the uniform traffic conditions under various loads using simulations. We compared the performance of 2-class priority mechanism against the single priority one, by gathering metrics 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. We found that the use of asymmetric-sized buffered systems leads to better exploitation of network capacity, while the increments in delays can be tolerated. The goal of this paper is to help network designers in performance prediction before actual network implementation and in understanding the impact of each parameter factor.**

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

I. INTRODUCTION

Multistage Interconnection Networks (MINs) with crossbar Switching Elements (SEs) are frequently proposed for interconnecting processors and memory modules in parallel multiprocessor systems. MINs have been recently identified as an efficient interconnection network for communication structures such as gigabit Ethernet switches, terabit routers, and ATM switches. A significant advantage of MINs is their good performance and low cost/performance ratio. A significant advantage of MIN-type interconnection systems is their ability to route multiple communication tasks concurrently. MINs with the Banyan [1] property as Omega Networks [7], Delta Networks [6], and Generalized Cube Networks [8] are proposed to connect a large number of processors to establish a multiprocessor system; they have also received considerable interest in the development of packet-switched networks. Non-Banyan MINs, are in general, more expensive than Banyan networks and more complex to control.

During the last decades, much research has targeted the investigation of parallel and distributed systems' performance, particularly in the area of networks and communications. In order to evaluate their performance different methods have been used such as Markov chains, queuing theory, Petri nets and simulation experiments.

Single priority queuing systems can be found in a large number of related articles. For example, [2, 3, 4] study the *throughput* and system *delay* of a MIN assuming the SEs have a single input buffer, whereas the performance of a finite-buffered MINs is studied in works such as [5]. In the industry domain, Cisco has built its new CRS-1 router [15, 16] as a multistage switch fabric. The switch fabric that provides the communications path between line cards is 3-stage, self-routed architecture.

A recent development in the MIN domain is the introduction of dual priority (or 2-class) queuing systems, which are able to offer different quality-of-service parameters to packets that have different priorities. Packet priority is a common issue in networks, arising when some packets need to be offered better quality of service than others. Packets with real-time requirements (e.g. from streaming media) vs. non real-time packets (e.g. file transfer), and out-of-band data vs. ordinary TCP traffic [9] are two examples of such differentiations. There are already several commercial switches which accommodate traffic priority schemes, such as [17, 18]. These switches consist internally of single priority SEs and employ two priority queues for each input port, where packets are queued based on their priority level. The performance of dual priority MINs has not been adequately investigated insofar, 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 this paper, we focus on *asymmetric-sized* buffered SEs that natively supports 2-class routing traffic, aiming to improve the QoS offered to high-priority packets. However, most works use equal buffer queue sizes for all priority classes [13, 14], despite the fact that –typically– normal priority packets outnumber their high-priority counterparts. In this paper we introduce a variation of double-buffered SEs that uses asymmetric buffer sizes for packets of different priorities, aiming to better exploit the network hardware resources and capacity.

The remainder of this paper is organized as follows: in section 2 we briefly analyze a Delta Network that natively supports 2-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 2-Class 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 [1] 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, individual queues have added for both high and low priority packets. In this scheme, each SE has two transmission queues per link, with one queue dedicated to high priority packets and the other dedicated to low priority ones. During a single network cycle, the SE considers all its links, examining for each one of them firstly the high priority queue. If this is not empty, it transmits the first packet towards the next MIN stage; the low priority queue is checked only if the corresponding high priority queue is empty. Packets in all queues are transmitted in a first come, first served basis. In all cases, at most one packet per link (upper or lower) of a SE will be forwarded for each pair of high and low priority queues to the next stage. The priority of each packet is indicated through a priority bit in the packet header.

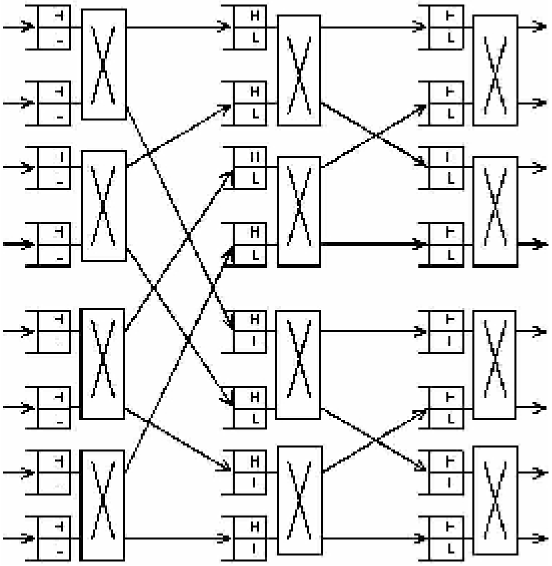

Figure 1. An 8 X 8 delta-2 network for 2-class routing traffic

An (N X N) MIN can be constructed by n=log$_c N$ stages of (cxc) SEs, where c is the degree of the SEs. 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.

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).

A Delta Network is assumed to operate under the following conditions:

- The network clock cycle consists of two phases. In the first phase, flow control information passes through the network from the last stage to the first one. In the second phase, packets flow from one stage to the next in accordance to the flow control information.
- The arrival process of each input of the network is a simple Bernoulli process, i.e. the probability that a packet arrives within a clock cycle is constant and the arrivals are independent of each other.
- Under the two-class priority mechanism, when applications 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) or from protocol intrinsics (e.g. TCP out-of-band/expedited data vs. normal connection data).
- 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 the destinations and each high/low priority queue uses a FIFO policy for all output ports.
- When two packets at a stage contend for a buffer at the next stage and there is no adequate free space for both of them to be stored (i.e. only one buffer position is available at the next stage), there is a conflict. Conflict resolution in a single-priority mechanism operates under the following scheme: one packet will be accepted at random and the other will be blocked by means of upstream control signals. Under the 2-class priority scheme, the conflict resolution procedure takes into account the packet priority: if one of the received packets is a high-priority one and the other is a low priority packet, the high-priority packet will be maintained and the low-priority one will be blocked by means of upstream control signals; if both packets have the same priority, one packet is chosen randomly to be stored in the buffer whereas the other packet is

blocked. The priority of each packet is indicated through a priority bit in the packet header, thus it suffices for the SE to read the header in order to make a decision on which packet to store and which to drop.

- All SEs have deterministic service time.
- Finally, all packets in input ports contain both the data to be transferred and the routing tag. In order to achieve synchronously operating SEs, the MIN is internally clocked. As soon as packets reach a destination port they are removed from the MIN, so, packets cannot be blocked at the last stage.

III. Performance Evaluation Methodology

In order to evaluate the performance of a (*N X N*) Delta Network with $n=\log_c N$ intermediate stages of (*c*x*c*) SEs, we use the following metrics. Let *T* be a relatively large time period divided into *u* discrete time intervals ($\tau_1, \tau_2, ..., \tau_u$).

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

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

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

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

$$Th = \frac{Th_{avg}}{N} \tag{2}$$

and reflects how effectively the network capacity is used.

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

$$RTh(h) = \frac{Th(h)}{\lambda_h} \tag{3}$$

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

$$RTh(l) = \frac{Th(l)}{\lambda_l} \tag{4}$$

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

$$D_{avg} = \lim_{u \to \infty} \frac{\sum_{i=1}^{n(u)} t_d(i)}{n(u)} \tag{5}$$

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

We consider $t_d(i) = t_w(i) + t_{tr}(i)$ where $t_w(i)$ denotes the total queuing delay for i^{th} packet waiting at each stage for the availability of an empty buffer at the next stage queue of the network. The second term $t_{tr}(i)$ denotes the total transmission delay for i^{th} packet at each stage of the network, that is just $n*nc$, where *n* is the number of stages and *nc* is the network cycle.

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

$$D = \frac{D_{avg}}{n * nc} \tag{6}$$

- *Universal performance (U)* is defined by a relation involving the two major above normalized factors, *D* and *Th:* the performance of a Delta Network 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 , *U* can be expressed by

$$U = \sqrt{D^2 + \frac{1}{Th^2}} \tag{7}$$

It is obvious that, when the *packet delay* factor becomes smaller or/and *throughput* factor becomes larger the *universal performance* factor *(U)* becomes smaller. Consequently, as the *universal performance* factor *(U)* becomes smaller, the performance of Delta Network 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 (7) can be replaced by:

$$U = \sqrt{\left(\frac{D - D^{\min}}{D^{\min}}\right)^2 + \left(\frac{Th^{\max} - Th}{Th}\right)^2} \tag{8}$$

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 (7), when the *universal performance* factor *U*, as computed by equation (8) is close to zero, the performance a Delta Network is considered optimal whereas, when the value of *U* increases, its performance deteriorates. Finally, taking into account that the values of both *delay* and *throughput* appearing in equation (8) are normalized, $D^{\min} = Th^{\max} = 1$, thus the equation can be simplified to:

$$U = \sqrt{(D-1)^2 + \left(\frac{1 - Th}{Th}\right)^2} \tag{9}$$

Finally, we list the major parameters affecting the performance of a 2-class priority Delta Network.

- *Buffer size* (b_h) of a high priority queue is the maximum number of packets that the corresponding input buffer of a SE can hold. In this paper we consider a finite-buffered Delta Network, where (b_h=1, 2).
- *Buffer size* (b_l) of a low priority queue is the maximum number of packets that the corresponding input buffer of a SE can hold. In this paper we consider a finite-buffered Delta Network, where (b_l=2, 3).

  We note here that the particular *buffer sizes* have been chosen since they have been reported [13] to provide optimal overall network performance: indeed, [13] documents that for smaller *buffer sizes* (1) the *network throughput* drops due to high *blocking probabilities*, whereas for higher *buffer sizes* (4 and 8) *packet delay* increases significantly (and the SE hardware cost also raises).
- *Offered load (λ)* is the steady-state fixed probability of arriving packets at each queue on inputs. In our simulation the λ is assumed to be λ = 0.1, 0.2… 0.9, 1. 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*)*, can be expressed by $r_h = \lambda_h/\lambda$. In our study r$_h$ is assumed to be r_h =0.20.
- *Network size n*, where n=log$_2$$N$, is the number of stages of an (N X N) Delta Network. In our simulation n is assumed to be n=10.

IV. Simulation and Performance Results

The performance of MINs is usually determined by modeling, using simulation [10] or mathematical methods [11]. In this paper we evaluated the network performance using simulation experiments due to the complexity of the model. For this purpose we developed a special simulator by C++ capable of handling 2-class priority Delta Networks. The simulator has several parameters such as the *buffer-length* of high and low priority queues respectively, the *number* of input and output ports, the *number* of stages, the *offered load*, and the *ratio* of high priority packets. The simulation was performed at packet level, assuming fixed-length packets transmitted in equal-length time slots, where the slot was the time required to forward a packet from one stage to the next. Each SE was modeled by four non-shared buffer queues, the first two for high priority packets, and the other two for low priority ones. Buffer operation was based on FCFS principle. The contention between two packets was solved randomly, but when a 2-class priority mechanism was used, the high priority packets had precedence over the low priority ones, where contentions were resolved by favoring the packet transmitted from the queue in which the high priority packets were stored in.

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

A. Single-buffered MINs and related work

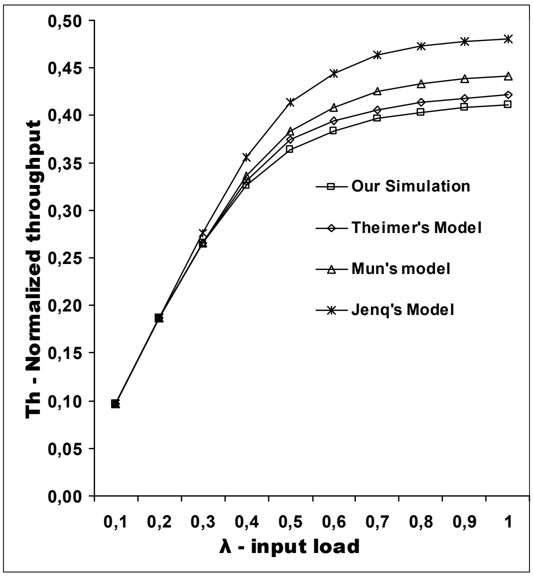

Figure 2. Normalized throughput of a single priority, single-buffered, 6-stage MIN

Figure 2 depicts the *normalized throughput* of a single-buffered 6-stage MIN versus the *offered load* for the three classical models [2, 3, 4] and our 2-class priority simulator in the marginal cases, when r_h=0 or r_h=1. It is obvious that all models are very accurate only at low *offered loads*, but the accuracy of the first two models seems to be reduced at both moderate and high traffic conditions. According to fig. 2 the accuracy of Jenq's model is insufficient, when the *offered load* is high. The reason is that many packets are blocked mainly at the first stages of the MIN at higher traffic rates. It is also worth noting that the accuracy of Mun's model was improved considerably by introducing a "blocked" state. Moreover, the accuracy of Theimer's model was further improved by considering the dependencies between the upper and low buffers of each SE. Our simulation was tested by comparing the results of the Theimer's model with those obtained by our simulation experiments which were found to be in a close agreement (differences were less than 1%).

Finally, figure 3 represents the *total normalized throughput* of a dual priority 6-stage MIN versus the *ratio of high priority packets* under full load *input traffic* conditions (λ=1); the two curves in this figure show the measurements obtained from Shabtai's Model reported in [12] and our model for the case of single-buffered MINs (b_h=b_l=1). We used these measurement to validate our simulator for the case of multi-priority MINs and found the two measurement sets to be in close agreement (maximum difference is 3.8%)

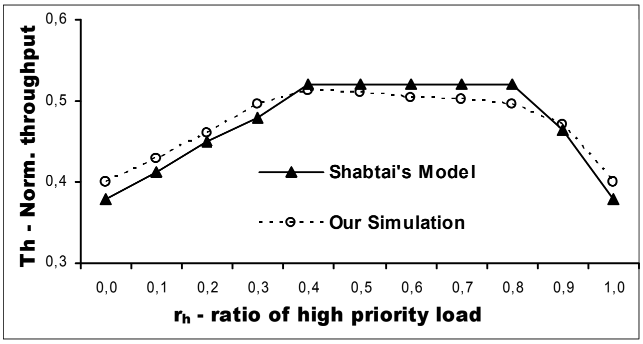

Figure 3. Total normalized throughput of a dual priority, single-buffered, 6-stage MIN

B. *Finite-buffered MINs with asymmetric-sized buffer queues*

In this paper we introduce a variation of double-buffered SEs that uses *asymmetric buffer sizes* in order to offer different quality-of-service parameters to packets that have different priorities, while providing in parallel optimal overall network performance.

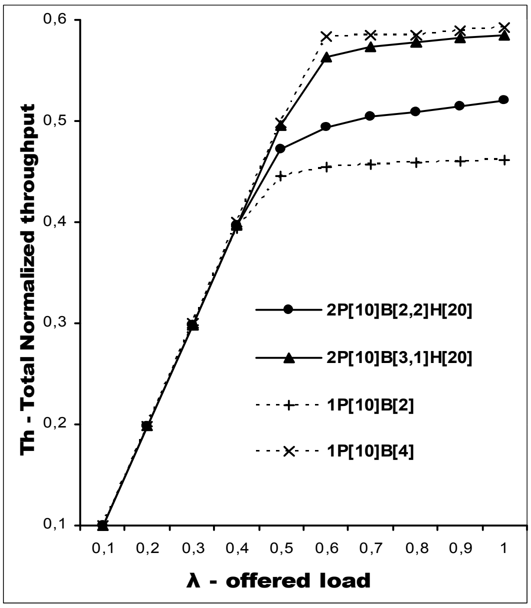

Figure 4. Total normalized throughput vs. offered load

In figure 4, curves 1P[10]B[*b*] depict the *normalized throughput* of a 10-stage Delta Network, under a single priority mechanism, when the *buffer-length* is *b*=2,4. Similarly, curves 2P[10]B[b_l,b_h]H[20] show the *total normalized throughput* of a 10-stage Delta Network, under a 2-class priority mechanism, when the *buffer-length* of low and high priority packets is b_l =2,3 and b_h=2,1 respectively, and the probability of high priority packet appearance is 20% . According to this figure the gain for *total normalized throughput* of a double-buffered Delta Network, employing a 2-class priority mechanism (curve 2P[10]B[2,2]H[20]) vs. the corresponding single priority one (curve 1P[10]B[2]) is 12.6%, under full traffic load. Considering that the rate of high priority packets is relatively low, and configuring thus a asymmetric *buffer-sized* system (curve 2P[10]B[3,1]H[20]) the *total normalized throughput* is further improved 14.1%, approaching that of a single priority mechanism, when the *buffer-length* is *b*=4, where all buffers serve all packets.

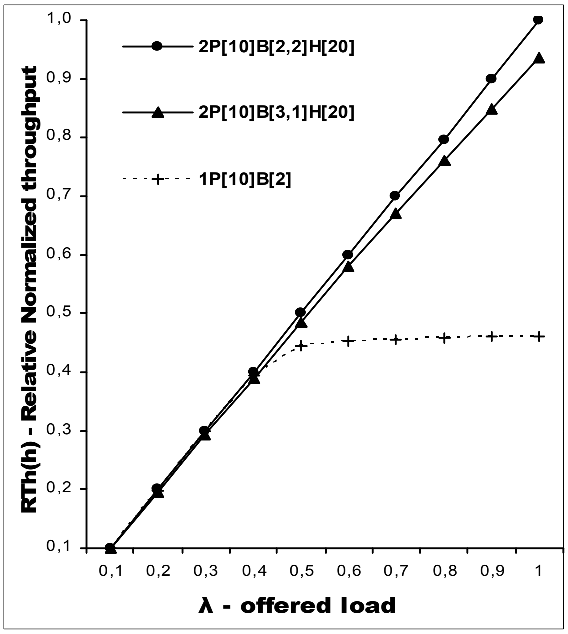

Figure 5. Relative norm. throughput of high priority packets vs. offered load

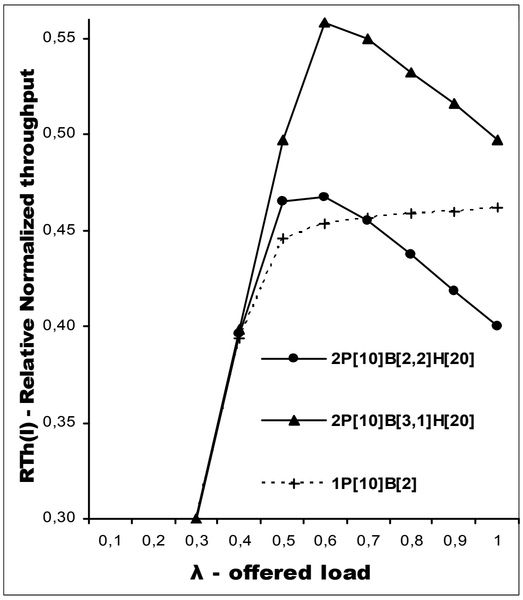

Figure 6. Relative norm. throughput of low priority packets vs. offered load

Figures 5 and 6 depict the *relative normalized throughput* of high and low priority packets respectively. According to figure 5 both curves employing the 2-class priority mechanism approach the optimal value $Th_{max}=1$ of this performance factor. It is obvious that, when the setup of *buffer-length* for high priority packets is $b_h=2$ (curve 2P[10]B[2,2]H[20]), the *relative normalized throughput* appears further improved, but the gains are marginal. Figure 6 presents the case of low-priority packet *throughput*; in this figure we can observe that the *relative normalized throughput* of low priority packets is considerably better, when the setup of *buffer-length* for high priority packets is $b_h=1$ (curve 2P[10]B[3,1]H[20]), as compared to the case of having equal-size buffers (curve 2P[10]B[2,2]H[20]), for high and low priority packets. The performance difference between the two setups is approximately 20% for medium and high network loads ($\lambda \geq 0.5$). We can also observe that the *asymmetric-sized* buffer setup offers superior service to the low-priority packets as compared to the single-priority scheme, mainly owing to the one additional buffer position available in the asymmetric setup to packets of this class. The performance improvement appears for medium and high network loads ($\lambda \geq 0.5$) and ranges from 8% to 21%.

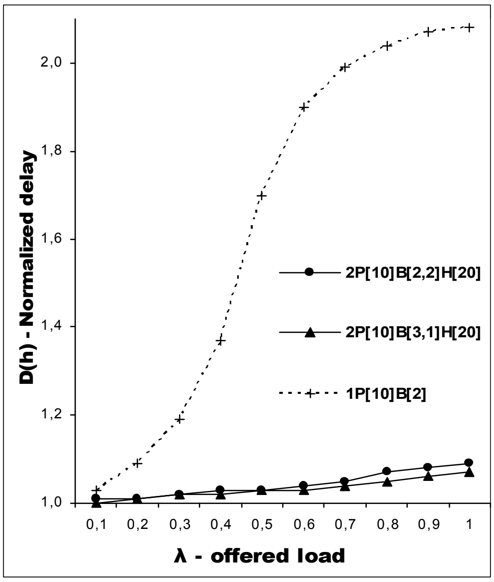

Figure 7. Normalized delay of high priority packets vs. offered load

Figures 7 and 8 present the findings for the *normalized delay* performance metric. In figure 7 we can observe that both 2-priority schemes (i.e. the *equal-sized* buffer and the *asymmetric-sized* buffer scheme) have a clear edge over the single-priority mechanism, which ranges from 18% at 30% load to over 96% at full load. The difference however between the performance of the *equal-sized* buffer scheme and the *asymmetric-sized* buffer scheme is very small, less than 4% in all cases. Conversely, when low priority packets are considered (figure 8), the *equal-sized* buffer scheme is found to have delays close to the single-priority scheme, with the worst case being a deterioration of 6.7% at *offered load* $\lambda = 1$. In the *asymmetric-sized* buffer setup however, the deterioration is considerable, especially at high loads (13% at $\lambda = 0.6$ rising up to 24.4% as compared to the *equal-sized* buffer setup at $\lambda = 1$).

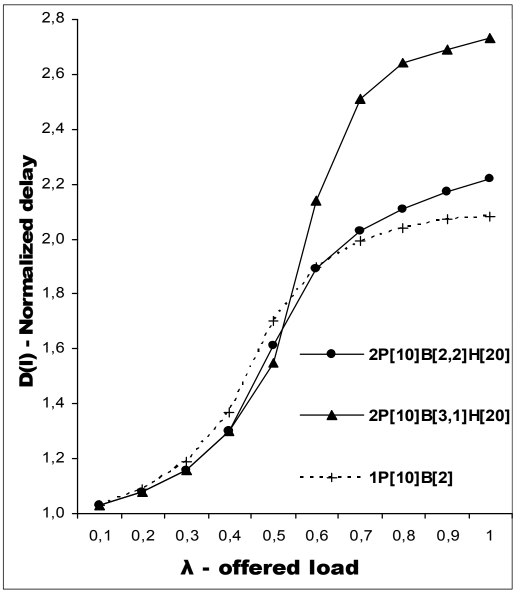

Figure 8. Normalized delay of low priority packets vs. offered load

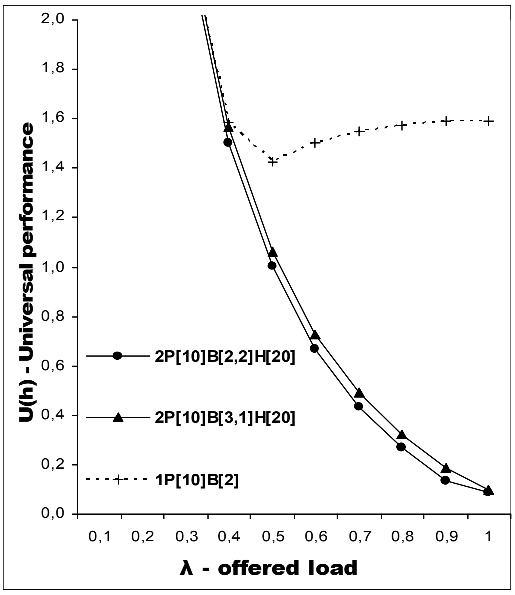

Figure 9. Universal performance of high priority packets vs. offered load

Figures 9 and 10 depict the behavior of the *universal performance factor* metric for high- and low-priority packets, respectively, in correlation to the *offered load*. We can observe in figure 9 that when the load of the network is relatively low ($\lambda \leq 0.4$), all configurations have identical performance; however, when the *network load* increases, the overall performance of the single-priority configuration quickly deteriorates, as compared to the setups supporting two priorities. The *asymmetric-sized* buffer configuration shows almost identical performance to the *equal-size* buffer configuration in this case, and both these performances are close to the optimal one.

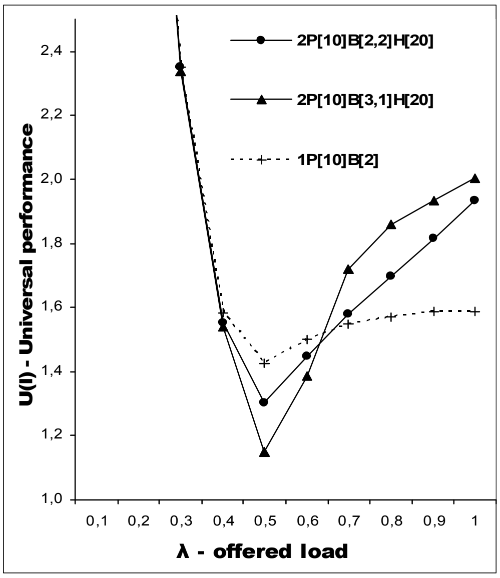

Figure 10. Universal performance of low priority packets vs. offered load

Regarding low-priority packets, again the overall performance of all configurations is identical for light *network loads* ($\lambda \leq 0.4$), Beyond this point, the single-priority setup exhibits the most stable behaviour, with the value of the *universal performance factor* for low-priority packets *U(l)* being close to 1.5; the single-priority setup has a clear advantage over the dual-priority schemes for *offered loads* $\lambda \geq 0.7$.

The configurations supporting two priorities exhibit a wider performance fluctuation, with the *asymmetric-sized* buffer configuration having a performance edge for *network loads* λ between 0.5 and 0.6, while for network loads $\lambda \geq 0.7$ the performance advantage moves to the *equal-size* buffer configuration side, not exceeding though 5.5% in any case.

V. CONCLUSIONS

In this paper we have introduced an *asymmetric buffer size* configuration for MINs supporting two packet priority classes and compared its performance against both the single-priority scheme and the typical, *equal-sized buffer* configuration of two packet priority classes MINs 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, when comparing the *asymmetric buffer size* configuration against its *equal-sized buffer* counterpart, we found that the former provides better overall *throughput* and significantly better low-priority packet *throughput* and *delay*; for high-priority packets on the other hand, the performance of the two schemes is almost identical, with the *equal-sized buffer* scheme having a small edge.

The *asymmetric-sized* buffer configuration achieves these performance benefits because it better matches buffer allocation to the shape of network traffic. Future work will focus on examining other load configurations, including hot-spot and burst loads, as well as different high/low priority ratios. The introduction of an adaptive scheme, altering buffer allocation to different priority classes according to current traffic load and high/low priority ratios will be investigated as well.

REFERENCES

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

[2] Y.-C.Jenq. "Performance analysis of a packet switch based on single-buffered banyan network", *IEEE Journal Selected Areas of communications* , pp. 1014-1021, 1983.

[3] H. Mun and H.Y. Youn. "Performance analysis of finite buffered multistage interconnection networks", *IEEE Tranactions.on Computers,* pp. 153-161, 1994.

[4] 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.

[5] 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.

[6] J.H. Patel. "Processor-memory interconnections for mutliprocessors", *Proceedings 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 Transactions on Computers*,. 31(4)5, pp. 443-454, May 1982

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

[10] 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.

[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", *Proceedings of the IEEE-sponsored International Joint Conference on Telecommunications and Networking 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", *Proceedings of the 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07)*, 2007

[15] http://www.cisco.com/application/pdf/en/us/guest/products/ps5763/c1031/cdccont_0900aecd800f8118.pdf.

[16] http://newsroom.cisco.com/dlls/2004/next_generation_networks_and_the_cisco_carrier_routing_system_overview.pdf.

[17] D-Link. DES-3250TG 10/100Mbps managed switch http://www.dlink.co.uk /DES-3250TG.htm.

[18] Intel Corporation. Intel Express 460T standalone switch. http://www.intel.com/support/express/switches/460/30281.htm.