

The *Dual-Banyan* (DB) Switch: A High-Performance Buffered-Banyan ATM Switch[†]

Christos Koliás and Leonard Kleinrock
{ck, lk}@cs.ucla.edu

Department of Computer Science
University of California at Los Angeles
Los Angeles, CA 90095-1596
tel.: +1 310 825-1563 fax: +1 310 825-2273

Abstract

Multistage Interconnection Networks (MINs) are very popular in ATM switching since they can achieve high-performance switching and are easy to implement and expand due to their modular design. In this paper we present and describe in detail a high-performance buffered-banyan switch which encompasses multiple input-queueing as its buffering strategy. We call this switching architecture Dual-Banyan switch. Simulation results are given to demonstrate its throughput, mean waiting time and cell-loss performance considering different switch and buffer sizes. We further compare it to the simple, single-queue buffered banyan network, assuming, for reasons of fairness, the same total buffer capacity and with respect to uniform and non-uniform traffic patterns.

1 Introduction

Multistage Interconnection Networks (MINs) constitute a large class of ATM switching systems that are widely used in today's internetworking. The basis for MINs was set back in the 50's and 60's with the deployment of the circuit-switched telephony. The original goal was to develop an internally nonblocking, multistage switch that would result in fewer crosspoints than an ordinary single-stage (i.e., crossbar) switch without, however, sacrificing any performance. MINs have been proven very useful and practical in the area of parallel computers and supercomputers for interconnecting memory modules and processors. This has resulted in proposals and studies of a plethora of architectures [1]. Advances in VLSI technology have facilitated the construction of large and high-capacity switch fabrics and at the same time have made possible for minimizing the overall cost.

Two of the most well-known types of Multistage Networks are the *Banyan* and the *Delta* networks [9] (Delta networks are a subclass of Banyan networks). Banyan networks are characterized by the fact that there is a single path between any

input-output pair. The basic switching element is usually a 2×2 crossbar switch and consequently an $N \times N$ Banyan switch consists of $\log_2 N$ stages and a total of $\frac{2}{N} \log_2 N$ switching elements. Banyan networks suffer from internal blocking, when two packets (or cells in the context of ATM switching) try to access the same internal link, even though they carry different final output port destinations. In addition, as it is common in any input-buffered switch, there can be output port contention causing output blocking.

There exist several alternatives for reducing or eliminating blocking [1]:

- Use buffers in the internal switching elements. As in crossbar switches, buffers can be placed at the inputs of the switching element, at its outputs or a shared (non-FIFO) buffering scheme can be used.
- Increase the speed at which the internal interconnection links operate.
- Implement a feedback or backpressure mechanism to schedule the switching of those packets that can cause blocking ahead.
- Introduce multiple MINs in parallel that will allow alternate paths between inputs and outputs.
- Apply a sorting network in front of the Banyan network that rearranges incoming packets in an increasing order (according to their output destinations) and guarantees no internal blocking within the banyan network. The Batcher network [5] is known to be such a sorting network and the resulting combination of this Batcher-Banyan network was first proposed in the design of the Starlite switch [4].

Due to their simple structure larger banyan MINs can be built in a modular fashion by smaller ones, which requires only minimal modifications and simplifies their VLSI and software implementation. Depending on whether the interconnection network can have buffers or not, banyan networks are distinguished into buffered and unbuffered [8].

In this paper we describe a buffered-banyan network which consists of 2×2 crossbar switches that employ multiple input-queueing [7] to store and switch the cells. We show that the achieved performance in terms of throughput and average delay

[†]This work was supported by DARPA/ITO under contract DABT63-93-C-0055, *The Distributed Supercomputer Supernet - A Multi Service Optical Intelligent Network*. The first author was also supported by the Lillian Voudouri Foundation (Greece) Fellowship. Web URL: <http://millennium.cs.ucla.edu/~kolia>

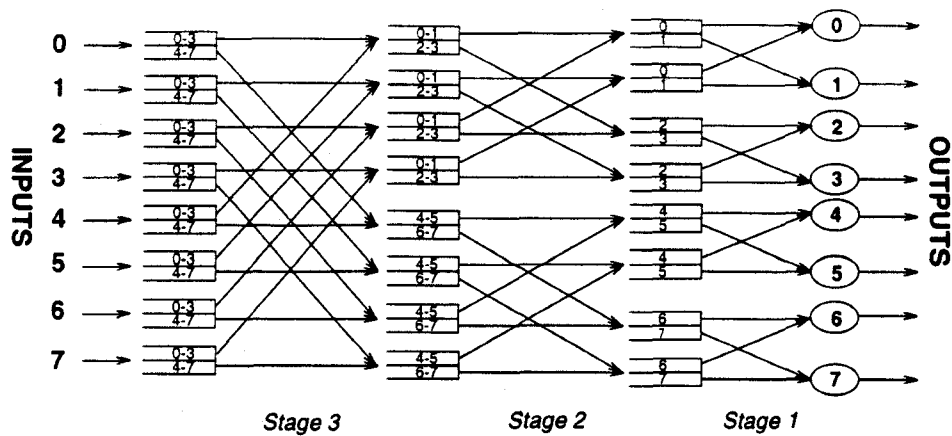


Figure 1: Schematic view of buffering and switching in a 3-stage MIN using dual queues at the inputs of every stage.

can be substantially improved compared to the simple banyan network [5].

Our results are based on a simulation approach where we considered networks of various parameters in terms of size (number of ports) and (total) buffer capacity. In the next section we introduce the *Dual Banyan* network and describe its main features. The performance evaluation of the Dual-Banyan network is discussed in section 3 and it is followed, in section 4, by a throughput and mean waiting time comparison to the ordinary buffered banyan network (with single queues), based on buffer management. Section 5 deals with the non-uniform traffic case, where we allow traffic to be directed with higher probability to particular output ports. Next, we briefly give some extensions, enhancements and ideas regarding the Dual-Banyan network as future work. We conclude by summarizing our main results and contributions.

2 The *Dual-Banyan* (DB) Multistage Switch

2.1 Architecture Description

The idea behind the proposed multistage network is that an incoming cell is stored, according to its intended output port, at either of the two queues (an upper queue or a lower queue) located at each input link. For that reason output ports are assumed to be partitioned into two groups, the upper half and the lower half group. We call this type of multistage interconnection network the *Dual Banyan* (DB) switch. Figure 1 depicts the concept of dual input-queueing in this type of multistage switch by showing an 8×8 DB network. This type of Banyan network is different than any parallel [3] or replicated [2] Banyan multistage networks.

More specifically, we consider a k -stage $N \times N$ MIN, that is with N input and N outputs (numbered from 0 to $N - 1$, and $N = 2^k$), where the k -th stage is the entry stage. A cell entering the j -th stage ($1 \leq j \leq k$) has one of the $N2^{j-k}$ output ports as its potential destination address. If it is destined to the upper (or lower) half of these outputs it is forwarded to the upper (or lower, respectively) queue (see figure 1).

The DB switch is based on a 2×2 crossbar switch that uses dual queues, at each port, to store the incoming cells. This type

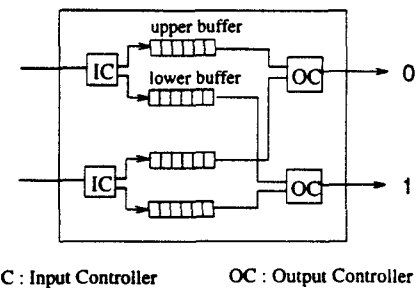


Figure 2: A 2×2 Dual-Banyan (DB) switching element.

of switching element (SE), which has a total of four queues, is shown in Figure 2. Queues inside the crossbar switch are FIFO queues and there are no additional queues at the output ports of the multistage switch.

As in multiple queueing switches [7] the existence of dual queues at each input helps to alleviate the *head-of-line* (HOL) blocking and to distribute the traffic load. It can be easily verified that the DB switch is a buffered, banyan-based network, as it has the following properties [12]:

1. The switch consists of $k = \log_2 N$ stages, and each switching element is a 2×2 multiple input-queueing crossbar switch.
2. There is a unique path from any input port to any output port.
3. It is an internally blocking switch, as two cells can attempt to access the same output link.
4. Due to its modular structure it is scalable.

Figure 3 shows the actual structure of the DB switch where the internal link connections have exactly the same topology as in an ordinary, single-queue Banyan switch. Inputs of the network are considered the inputs of the SEs at the entry (k -th) stage, while outputs of the network refer to the outputs of the SEs at the exit (1st) stage.

The multistage network is governed by a central clock, on a synchronous basis. The time unit, referred as the time slot, is defined as the time required for a cell to get routed from one stage to the next, it is constant (due to the fixed cell length) and its actual duration depends on the SE's fabric speed. Therefore.

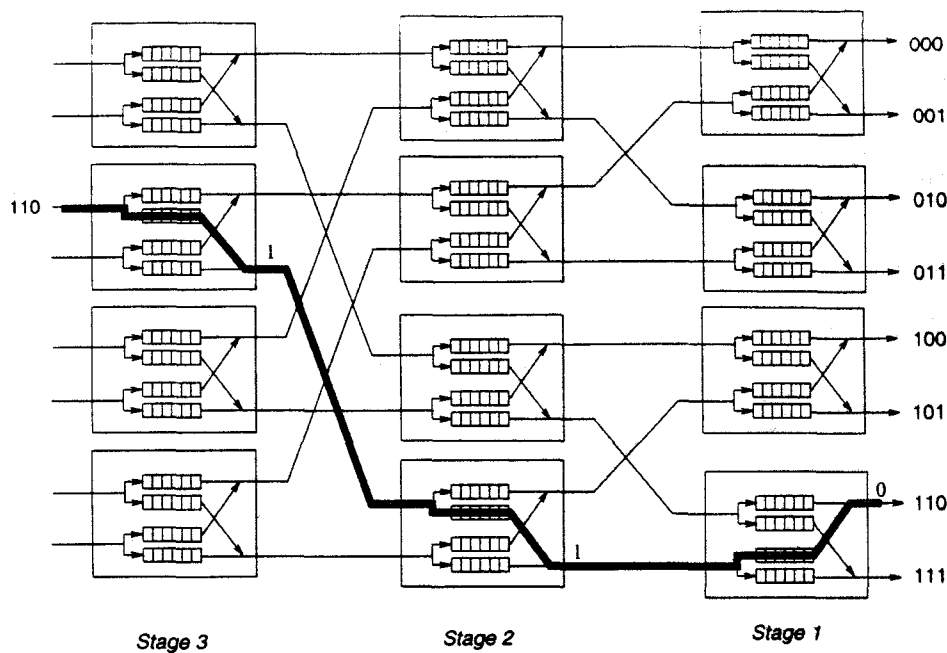


Figure 3: Illustration of self-routing in an 8x8 Dual-Banyan (*DB*) Multistage Switch.

a cell entering a k -stage network will need at least k time slots until it reaches its desired outgoing trunk. We assume that cell arrivals occur at the very end of the time slot while cells get switched immediately following the arrival phase (no queueing).

As far as arbitration in each SE is concerned, for selecting the HOL cell to be forwarded to the next stage or final output port, a round-robin scheme is applied. A cyclic order per output port is maintained during the arbitration (at the beginning of each time slot) and that order interchanges in every time slot for fairness purposes: if the queue associated with the upper input is arbitrated first the queue in the lower input is then polled first in the successive time slot. The two queues associated with the same input operate independently. Also, in order to improve performance, if the first FIFO does not transmit a cell (either it has no cell or the destination queue is full) then the second one (from the other input) is considered for transmission. Thus, an input can send up to two cells (one from each of its FIFO queues) during the same time slot. Only one of the two queues connected to the same outgoing link can send a cell, as we do not allow any speed-up of links. If there is a conflict, the other cell is buffered and awaits its turn to be switched in the next time slot. It is clear that the arbitration algorithm designates two independent and concurrent arbitration phases, one for output "0" and one for output "1" (see figure 2). Essentially each SE operates as a 2×2 crossbar multiple input-queueing switch (see [7], policy B). Note that the resulting switching architecture turns out to be the same as that described in [10] where crossbar switches are implemented with the buffers inside them.

Other selection policies can be also considered such as the random selection policy in which a queue is randomly chosen to send its HOL cell. Another alternative is to select a cell from the longest queue; this is actually a more favorable strategy as it tends to avoid overflowing some queues while other queues are less full (especially under a non-uniform traffic patterns).

2.2 Switch Characteristics

A distinguishing feature of the SE used to implement the *DB* switch is that due to the presence of dual-queues the arbitration process regarding the upper output port ("0") in the SE is independent than the one for the lower ("1") port. That enables the SE to switch out two cells (per time slot), i.e., in the case where all four queues have at least one cell each. This fact can potentially improve the overall performance of the multistage network.

Another significant characteristic is that the switching and forwarding functions are separated, as they occur at different places inside the crossbar switch. Input Controllers (ICs) (e.g., demultiplexers) are placed at the inputs of the queues to determine which queue the incoming cell should join; therefore the switching of cells is performed before storing them. Also, Output Controllers (OCs) (e.g., multiplexers) are located at the outputs of each element in order to arbitrate between the two queues that access the same outgoing link. Forwarding to the outgoing links is done in an independent fashion, once the arbitration and selection phase is completed. Broadcasting is straightforward to implement by simply allowing the input controllers to copy the cell and store both of them to the queues.

It is easy to verify that the *DB* possesses the *self-routing*¹ property, which then characterizes the *DB* network as actually a Delta network. A cell gets switched through the stages based on the binary representation of its intended output port number, known as the address tag. For instance, if the address tag of a cell is the sequence $d_k d_{k-1} \dots d_1$ then the cell is routed (and buffered accordingly) through the j -th stage based on the d_j digit. Figure 3 demonstrates the self-routing property of the *DB* network, where if "0" is read in the cell's routing tag it joins the upper queue while if "1" is read it joins the lower queue.

¹ Also known as *digit-controlled routing*.

3 Performance Results

The banyan network we described provides buffers in every switching element to alleviate some of the internal blocking. Since there are no output buffers output blocking is present due to output port contention. Both these types of blocking, namely internal and final output trunk blocking, cause a degradation of the switch's performance. Blocking becomes more severe as the switch size increases since potential conflicts have a higher probability of occurrence.

The performance evaluation carried is based on simulation results. We consider different queue sizes b and various network sizes N (number of input or output ports). Traffic at the inputs of the DB network is uniform; an incoming cell is destined equiprobably to any of the outputs of the network. Cell arrivals to each entry input port are independent and distributed according to the same Bernoulli process (balanced traffic) with parameter p , with the latter also denoting the offered traffic load.

A saturation situation occurs when the traffic load exceeds the achievable (maximum) throughput of the switch and can be exhibited at different network and queue sizes. However an arrival rate of $p=1.0$ always causes the switch to be saturated. At saturation, network transit delays and cell loss due to buffer overflow become very large.

Two different types of SEs can be considered for implementing a MIN [13]: a *discarding* crossbar, where a cell is dropped from the switch if there is not enough buffer space to store it and a *blocking* crossbar switch where backpressure is used as a flow control mechanism to prevent buffer overflow inside the switch². In the latter case cell loss can evidently occur only at the entry stage of the MIN in the case of insufficient storage capacity. In our simulation we examined the DB switch under both types of crossbar switches and concluded that both the resulting MINs, namely with blocking and discarding SEs, yield exactly the same throughput.

A multistage network composed of blocking SEs requires additional hardware in order for a crossbar to communicate and notify other switches of its state (i.e., full queues), which increases the complexity of the VLSI implementation. From that viewpoint discarding switches can be chosen instead, for their simplicity. Cells that are dropped from a discarding SE do not re-enter the DB switch, so they are assumed permanently lost.

Figure 4 shows the throughput of the DB switch for different queue sizes, b . We notice that the throughput in the case of $b=1$ deteriorates rapidly with the network size N while as we add more buffer space, throughput increases and becomes more stable. In fact there is a big "jump" as buffer size increases from $b=1$ to $b=2$, for very large N (e.g. $N=1024$). For $b=32$ we obtain a throughput approaching 100%. We also observe from the same figure and looking at the two top curves, that the infinite buffer case can be approximated by using finite buffers of size 32, as the two throughputs are indeed very close³.

We previously mentioned that as far as throughput is concerned, blocking and discarding MINs behave identically. However, as shown in Tables 1 and 2, considering blocking SEs can

² A cell at stage j is not forwarded to the next stage $j-1$ if the reception buffer is full.

³ Our results are, and in some cases considerably, different from those presented in [10].

result in a smaller cell loss (which in fact remains almost constant regardless of the switch size). This kind of behaviour is anticipated as the backpressure mechanism "restrains" any cells in previous stages, that would be otherwise dropped, indicating an optimal utilization of the queues.

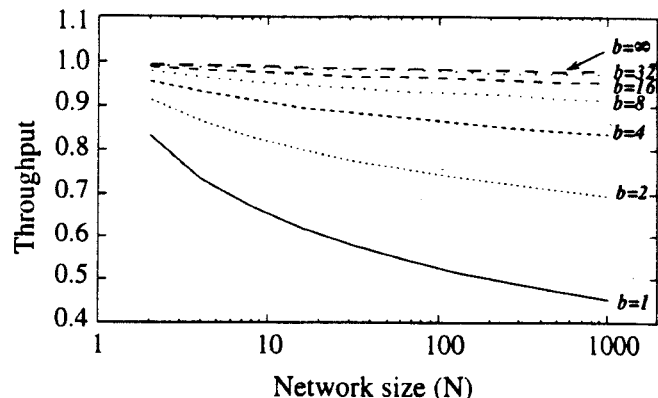


Figure 4: Throughput of the DB switch for different queue sizes b (simulation).

Network Size (N)	Cell Loss (at saturation)	
	2×2 blocking SE	2×2 discarding SE
4	$4.06 \cdot 10^{-2}$	$6.61 \cdot 10^{-2}$
8	$4.04 \cdot 10^{-2}$	$8.52 \cdot 10^{-2}$
64	$4.01 \cdot 10^{-2}$	$12.54 \cdot 10^{-2}$
512	$4.03 \cdot 10^{-2}$	$15.24 \cdot 10^{-2}$
1024	$4.01 \cdot 10^{-2}$	$15.96 \cdot 10^{-2}$

Table 1: Cell drop ratio (defined as the fraction of the cells entered the network that are dropped due to blocking) in the DB using blocking SEs vs. using discarding SEs, total buffer space per port $b=4$ (simulation).

Network Size (N)	Cell Loss (at saturation)	
	2×2 blocking SE	2×2 discarding SE
4	$1.99 \cdot 10^{-2}$	$3.20 \cdot 10^{-2}$
8	$2.03 \cdot 10^{-2}$	$4.23 \cdot 10^{-2}$
64	$2.01 \cdot 10^{-2}$	$6.34 \cdot 10^{-2}$
512	$2.00 \cdot 10^{-2}$	$7.78 \cdot 10^{-2}$
1024	$1.99 \cdot 10^{-2}$	$8.14 \cdot 10^{-2}$

Table 2: Cell drop ratio in the DB using blocking SEs vs. using discarding SEs, total buffer space per port $b=8$ (simulation).

4 The DB vs. the single-queue Buffered-Banyan network

In this section we evaluate the performance of the DB network compared to that of the single-queue Buffered-Banyan network. For our simulation runs we used the same set of assumptions

as previously described. Figures 5-8 demonstrate how the *DB* performs compared to the ordinary Banyan network, in terms of throughput, for various buffer sizes and under saturation. In order to make a fair comparison, the total buffer space for each input port is assumed to be the same for both Banyan networks; that is if queues are 8 cells long in the single-queue Banyan then they are 4 cells long in the *DB* network. The *DB* network considered is based on blocking SEs.

Figure 5 corresponds to the case where buffers in the *DB* switch can hold up to one cell and exhibits that it performs poorly compared to a Banyan network when the network grows to more than 5 stages. That indicates that internal blocking, due to insufficient buffer space, can have a detrimental effect on the throughput and consequently the overall performance of the network. Therefore, a *DB* network with single-cell queues is not an efficient and reasonable solution. However, as we move up in queue size the *DB* network behaves significantly better, where the absolute difference between the two throughputs, namely of the *DB* and the Banyan network, increases with the queue size (see figures 6-7). More noticeably, if $b = 32$ and 64 for the *DB* and the ordinary Banyan network respectively, the former one achieves a throughput of almost 100% with a lagging performance of 73% for the Banyan (see figure 8). The superior throughput performance of the *DB* network with $b > 1$ is attributed to the fact that the HOL impact inside each SE is mitigated when applying the dual input-queueing scheme. This beneficial effect propagates throughout the stages of the network. It is clear that providing more buffer space inside the SEs causes blocking (and consequently cell loss) due to buffer overflow to be lessened; however, as will we see, it increases the mean cell delay in the network. It is important to re-emphasize the fact that in the *DB* network a SE can send two cells in the course of a time slot, assuming, for instance, a heavy-load situation while in the single-queue Banyan output contention can limit the SE's, and thus the overall, throughput.

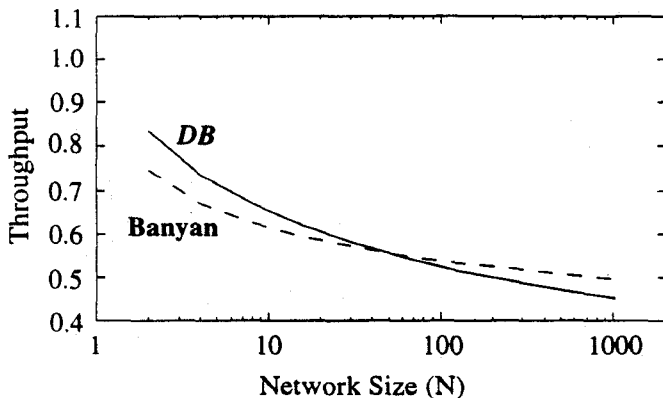


Figure 5: Throughput of the *DB* network vs. the single-queue Banyan network, with total buffer space $b=2$ (simulation).

Similar observations can be also drawn regarding the mean waiting times of the two networks. The mean waiting time is defined as the average total time spent in the queues (queueing time) in all the stages of the network. In order to get a measure of the mean total delay (mean transit time) we have to also take into account the total transmission delay (when a cell is advanced from one stage to the next and finally to its output port) which amounts to k slots for a k -stage Banyan.

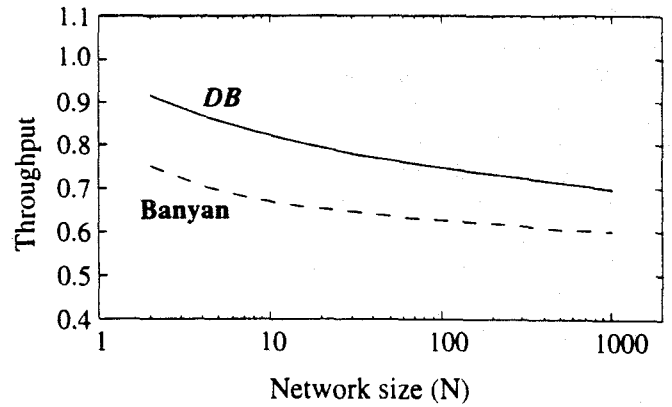


Figure 6: Throughput of the *DB* network vs. the single-queue Banyan network, with total buffer space $b=4$ (simulation).

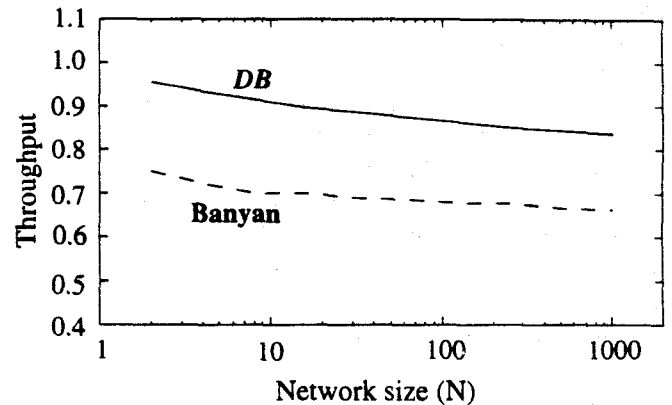


Figure 7: Throughput of the *DB* network vs. the single-queue Banyan network, with total buffer space $b=8$ (simulation).

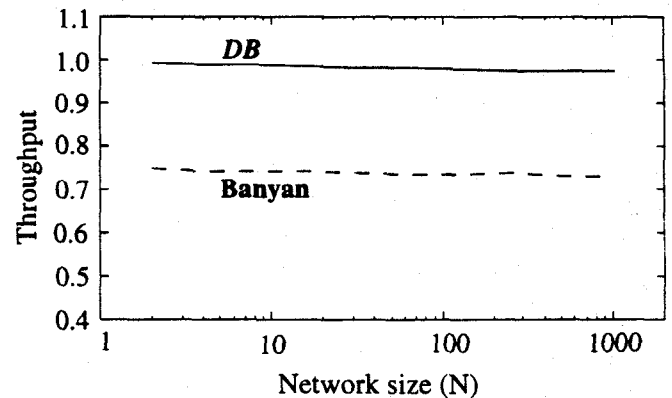


Figure 8: Throughput of the *DB* network vs. the single-queue Banyan network, with total buffer space $b=64$ (simulation).

Network Size $N \times N$	Buffer Space (per input port)							
	$b = 2$		$b = 4$		$b = 8$		$b = 64$	
	<i>DB</i>	Banyan	<i>DB</i>	Banyan	<i>DB</i>	Banyan	<i>DB</i>	Banyan
4×4	0.44	1.86	1.28	4.62	2.90	10.04	25.64	84.34
64×64	0.62	2.26	1.48	5.16	3.10	10.62	24.66	85.14
256×256	0.68	2.38	1.54	5.26	3.16	10.72	24.46	85.32
1024×1024	0.74	2.50	1.60	5.42	3.22	10.92	24.72	85.80

Table 3: Mean waiting times (in timeslots) for the *DB* and the single-queue Buffered-Banyan networks for various queue capacities (simulation).

Network Size $N \times N$	Buffer Space (per input port)					
	$b = 4$		$b = 8$		$b = 32$	
	<i>DB</i>	Banyan	<i>DB</i>	Banyan	<i>DB</i>	Banyan
64×64	0.38	0.25	0.38	0.25	0.38	0.25
256×256	0.36	0.23	0.37	0.23	0.37	0.23
1024×1024	0.36	0.23	0.37	0.23	0.37	0.23

Table 4: Throughput of the *DB* vs. the single-queue Banyan network, non-uniform traffic (simulation).

Table 3 contains simulation results for the mean waiting time of the *DB* and the single-queue Banyan network. We notice, from the same table, the advantage of using the *DB* network over the single-queue Banyan, which, once more, becomes more profound as buffer space increases. As expected, adding more buffer space causes delays to become longer (but, as mentioned earlier, reduces any potential cell loss).

5 Non-uniform traffic

We now extend our performance evaluation by examining the *DB* network and comparing it to the single-queue Banyan network, under non-uniform traffic. Any traffic pattern in which destination ports are not selected by the incoming cells with equal probability designates a non-uniform traffic pattern. It is known that non-uniform traffic can have a severe impact on the performance of any switch [6], since certain output ports can accept more cells than other ports. Such outputs, that have higher probability to be the destination of an incoming cell, are referred as *hot spots*. In addition to having hot spots among the outputs of the network, due to the nature of the non-uniform traffic it is probable that switches inside the network may exhibit severe overloaded situations as traffic passes through.

In our study we use an equally weighted mixture of uniform and non-uniform traffic. The latter is described by the following: the output port mapping for each incoming cell is determined by a binomial distribution [11], where for $i = 1, \dots, N-2$:

$$\begin{aligned}
 a_{i,j} &= Pr[\text{a cell arriving at the } i\text{-th network input is} \\
 &\quad \text{destined to the } j\text{-th network output}] \\
 &= \binom{N-1}{j} r_i^j (1-r_i)^{N-1-j}, \quad j = 0, \dots, N-1 \quad (1)
 \end{aligned}$$

and r_i is a probability associated with input i , $0 < i < N-1$. For this binomial distribution the maximum probability occurs

for $j = \lfloor (N-1)r_i \rfloor$. This allows us to choose $N-1-i$ to be the output for which the highest percentage of traffic arriving on i is directed (therefore it has the largest probability associated with input i), then we have

$$r_i = 1 - \frac{i}{N-1}, \quad i = 1, \dots, N-2. \quad (2)$$

For inputs 0 and $N-1$ the output address for $j = 0, \dots, N-1$ is given by a normalized Poisson-like distribution with rate r ,

$$a_{0,j} = \frac{r^{N-1-j}}{(N-1-j)!} \quad \text{and} \quad a_{N-1,j} = \frac{r^j}{j!} \quad (3)$$

A key point in adopting a binomial distribution for inputs $i = 1, \dots, N-2$, is that we still retain the balanced traffic effect in the *DB* network where both queues attached to each input port accept almost the same traffic load. However, this observation does not apply for inputs 0 and $N-1$, where, depending on the value of r , there is some bias introduced against the certain outputs, but as N grows this bias becomes negligible. Another advantage for using this type of binomial distribution is that we can obtain several hot spots (instead of just one or two), which in fact pertains to real-traffic scenarios as we can have multiple VC/VP connections all requesting the same output.

A value of $r=0.4$ was considered in our simulation runs; however, it should be noted that changing r has no significant impact in estimating the throughput of the system. Table 4 illustrates how both the *DB* and the Banyan networks behave under the resulting mixed traffic pattern, considering various buffer capacities⁴ and network sizes. A few comments regarding the results presented in Table 4, that apply - independently - to both networks: first, we notice that the throughput is not affected measurably by the network size (we saw that in the

⁴ $b=32$ means that queues are 16 cells long in the *DB* network and 32 cells long in the single-queue Banyan.

uniform case it deteriorates as N increases). Moreover we see that the throughput remains almost constant for the different buffer sizes shown in the same table. Both these observations lead us to the conclusion that the uneven destination port distribution has a greater impact than internal blocking due to buffer overflow or increase of the network size. Secondly, the improvement in throughput while using the *DB* network over the single-queue Banyan is more than 50% for this type of non-uniform traffic, definitely a considerable advantage for the *DB* network.

6 Future Work

We can further enhance the *DB* network by introducing multiple links to interconnect the queues at the different stages or by increasing the transmission speed of the internal links. For instance a speed-up⁵ of two would allow two cells to be switched from one stage to the next on the same link, in the same time slot; however output links retain the same speed as the inputs, so only up to one cell can be transmitted out on a output trunk. We anticipate that speeding up the internal links will have a beneficial effect on buffer performance (since blocking to buffer overflow will be reduced) and consequently to the overall performance of the network.

Another direct extension and generalization to our model would be to consider $d \times d$ crossbar switches as SEs, that will exercise multiple input-queueing, where, for instance, we can have d (or less) FIFO queues per input port.

7 Conclusion

Acknowledging the need for large, high-performance and cost-effective multistage switches we have described a Banyan-based network, the *DB* switch, which uses dual input queues in its switching element to buffer cells destined to different outputs. In fact we have considered two types of 2×2 crossbar switching elements, the blocking and the discarding switches, and briefly discussed their merits.

We have evaluated the *DB* switch by virtue of its throughput performance with respect to various network and buffer sizes. We have further assessed its performance by comparing it to a single-queue Banyan switch, based on throughput and mean waiting time results. We concluded that for the *DB* switch to be efficiently and practically functioning, it must have buffers that are at least as large as one cell. We also saw that increasing its queue size improves its throughput considerably in comparison with the ordinary Banyan. For instance increasing the total, per port, queue size from 4 to 8 leads to a throughput increase of 21% for the *DB* switch while only 11% for the banyan (see figures 6 and 7).

We recognize that we can improve a multistage network's performance at the expense of using additional hardware (e.g., C's in our scheme) and complexity, which of course makes such solutions more expensive. However, we feel that a compromise can be always pursued according and depending on the user's and/or network's needs and requirements.

⁵ A speed-up makes of course sense for networks with buffers larger than one ($b > 1$).

References

- [1] H. Ahmadi and W. Denzel, "A Survey of Modern High-Performance Switching Techniques", *IEEE Journal on Selected Areas in Communications* 7(7), pp. 1091-1103 (September 1989).
- [2] C. Catania and A. Pattavina, "Analysis of Replicated Banyan Networks with Input and Output Queueing for ATM Switching", in *Proceedings of ICC '96*, pp. 1685-1689, Dallas, Texas (June 1996).
- [3] C.-A. Fun and J. Silvester, "A New Parallel Banyan ATM Switch Architecture", in *Proceedings of the International Conference on Communications (ICC) '95*, Seattle, Washington, pp. 523-527 (June 1995).
- [4] A. Huang and S. Knauer, "Starlite: A Wideband Digital Switch", in *Proceedings of Globecom '84*, New Orleans, Louisiana, pp. 121-125 (November 1984).
- [5] J. Y. Hui, *Switching and Traffic Theory for Integrated Broadband Networks*, Kluwer Academic Publishers, Norwell, MA (1991).
- [6] H. S. Kim and A. Leon-Garcia, "Performance of Buffered Banyan Networks Under Nonuniform Traffic Pattern" *IEEE Transactions on Communications*, 38(5), pp. 648-658 (May 1990).
- [7] C. Koliass and L. Kleinrock, "Throughput Analysis of Multiple Input-Queueing in ATM Switches", *Broadband Communications*, L. Mason and A. Casaca (eds.) Chapman & Hall, London, U.K. (1996).
- [8] C. P. Kruskal and M. Snir, "The Performance of Multistage Interconnection Networks for Multiprocessors", *IEEE Transactions on Communications*, C-32(12), pp. 1091-1098 (December 1983).
- [9] M. Kumar and J. R. Jump, "Generalized delta networks", in *Proceedings of the 1983 IEEE International Conference on Parallel Processing*, Silver Spring, Maryland, pp. 10-18 (August 1983).
- [10] M. Kumar and J. R. Jump, "Performance Enhancement in Buffered Delta Networks Using Crossbar Switches and Multiple Links", *Journal of Parallel and Distributed Computing*, 1(1), pp. 81-103 (August 1984).
- [11] S.-Q. Li, "Nonuniform Traffic Analysis on a Nonblocking Space-Division Packet Switch", *IEEE Transactions on Communications*, 38(7), pp. 1085-1096 (July 1990).
- [12] R. Onvural, *Asynchronous Transfer Mode Networks: Performance Issues*, Artech House, Norwood, MA (1994).
- [13] Y. Tamir and G. L. Frazier, "Dynamically-Allocated Multi-Queue Buffers for VLSI Communication Switches", *IEEE Transactions on Computers* 41(6), pp. 725-737 (June 1992).