A SCALABLE FAMILY OF HIGH-SPEED SWITCH ARCHITECTURES

Mayez Al-Mouhamed

Computer Engineering Department
King Fahd University of petroleum & Minerals
Dhahran 31261, Kingdom of Saudi Arabia.
mayez@ccse.kfupm.edu.sa

ABSTRACT

In this paper we present a scalable and recursive class of banyan switching architectures called the *Shared-Tree Banyan Networks* (STBN). For ATM networks STBN can be engineered between two extremes: (1) a low-cost banyan with internal and external conflicts, or (2) a high-cost conflict-free fully-connected network with multiple outlets. STBN scalability is based on: (1) scalable concentrator bandwidth, and (2) controllable internal conflicts using path dilation. Scaling up the concentrator bandwidth leads to better utilization of the internal resources in blocking structures. Increasing path dilation increases service rate and cost. Evaluation shows that a small concentrator bandwidth combined with a moderate dilation degree produce a significant decrease in CLP by up to 10^{-3} fold the standard banyan. The STBN provides an effective tool to the scalability of banyan networks. It is very selective in bandwidth allocation by favoring higher-priority traffic which provides Q.o.S guarantees for selectively switching real-time traffic. To gain scalability, high-speed switching architectures can use the STBN as a basic banyan.
Keywords: ATM, banyan, dilation, service rate, switch.

1. INTRODUCTION

High-speed switching technology is based on extremely high-speed switching devices [1, 2, 3] with short fixed-length packets requiring minimal transmission delay and delay jitter and extremely low loss probability.

Banyan-based multistage networks are suitable for VLSI implementation because of their modularity, self routing, and low hardware complexity. One common approach used to increase throughput of banyans is to partition the input cells into conflict-free subsets so that the cells of each subset can be routed simultaneously without conflict in one single banyan. In many proposed switches [2], and [4] all input cells are issued to first banyan, successfully self-routed are retrieved, while all unsuccessful cells are re-issued to next banyan, and so on. This provides some sequential conflict resolution phases during which path reservations are processed in parallel within each banyan but sequentially iterated with respect to distinct banyans. Complex techniques are proposed to eliminate: (1) internal conflicts [4], (2) HOL blocking [5], or (3) reuse of idle inputs [3]..

We present a scalable and recursive class of banyan switching architectures called the *Shared-Tree Banyan Networks* (STBN). The scalability of the STBN is based on two factors: (1) scalable concentrator bandwidth, and (2) controllable internal conflicts using path dilation. Scaling up the concentrator bandwidth leads to better utilization of the internal resources in blocking structures. Increasing the path dilation degree reduces internal conflicts, which produces increase in service rate due to increase in throughput and decrease in path delay.

The organization of this paper is as follows. Section 2 we present a taxonomy of STBN. Section 3 presents the concentrator. The complexity analysis and analytical model are presented in Sections 4 and 5, respectively. Section 6 presents the evaluation and comparison. We conclude in Section 7.

2. SHARED-TREE BANYAN NETWORKS

The *Shared-Tree Banyan Networks* (STBNs) belongs to a class of dynamic, full access, single path, blocking multistage interconnection networks. It has two phases: 1) a *dilation phase* which is non-blocking, and 2) a *concentration phase* which is blocking. The architecture of STBN enables: (1) reducing internal blocking by increasing concentrator bandwidth (scaling-up), and (2) expanding the internal channel bandwidth of the dilation phase to reduce internal conflicts and to create multiple ports for each output channel. In general a d-dilated STBN has $N = 2^n$ inputs and $1{:}2^d$ dilation. There are d dilation stages and $n-d$ concentration stages where d always satisfies $0 \leq d \leq n$.

In the dilation phase the internal links are doubled by using Demultiplexers DMs, which are 1×2, at each of the d stages until reaching a 2^d-dilation at the dth stage. The stages are numbered from 1 to n. The header of each cell has an n-bit label and each bit is used for routing in a given stage. The DM is non-blocking because each request is allocated to its desired output. At full load, the dth stage has 2^{n+d} outputs which carry at most 2^n physical cells. Each cell is routed into one of the 2^d bundles, where each bundle has 2^n ports with at most 2^n cells. A butterfly permutation (β_s^{n+s}) is used to route 2^n DM upper output lines corresponding to a 0-routing bit from the

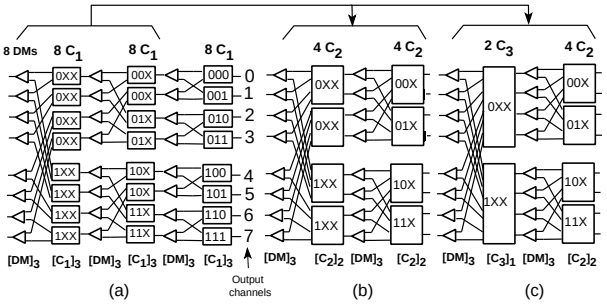

Figure 1. $STBN(3,0)$ with different blocking degrees

lines corresponding to a 1-routing bit (lower DM output). The butterfly permutation is used to group lines that carry requests with the same routing bit. Formally, the butterfly permutation used here is defined by

$$\beta_{d+s}^{n+d+1}(b_{n+d},\ldots,b_{n-s+1},b_{n-s},\ldots,b_0) =$$

$$(b_{n+d},\ldots,b_0,b_{n-s},\ldots,b_{n-s+1}) \qquad (1)$$

where $n+d+1$ is the number of bits, and $n-s+1$ is the position of the exchanged bit, and $(b_{n+d},\ldots,b_0)$ represents the line number.

The concentration phase has $n-d$ stages each has 2^{n+d} inputs and outputs. Consider the $(d+s)$th stage which is also the sth stage of concentration phase. As a result of routing through $d+s-1$ stages, the input of stage $(d+s)$ consists of 2^{d+s-1} bundles each has $2^{n-(s-1)}$ input lines. A set of 2^{n+d} DMs expands each bundle to 2^{n-s+2} lines. Grouping of all DM upper outputs and all DM lower outputs within each bundle of 2^{n-s+2} lines is done by using a permutation that will be described later. This subdivides each input bundle into an upper and lower bundles, thus making a total of 2^{d+s} output bundles each has 2^{n-s+1} lines. We introduce a blocking concentrator C_n having 2^n inputs and 2^{n-1} outputs. Ideally the concentrator function is to allocate its 2^{n-1} outputs to most prior 2^{n-1} inputs amongst its overall 2^n inputs.

Figure 1 shows an $STBN(3,0)$ with different blocking degrees. In Figure 1-(a) eight C_1 are used in each stage. To increase the concentrator bandwidth we may reduce the potential of internal conflicts in the $STBN(3,0)$ by replacing each set of $2\times C_1$ by $1\times C_2$ (Figure 1-(b)) or replacing each $4\times C_1$ by $1\times C_3$ (Figure 1-(c)) whenever the concentrator sets have the same routing tag. The bandwidth can be scaled up for each bundle where all the requests carry the same routing label. Similarly Figure 2 shows an $STBN(3,1)$ with different blocking degrees. In Figure 2-(a) we used $16\times C_1$ in second and third stage. To increase the concentrator bandwidth we may replace each set of $2\times C_1$ by $1\times C_2$ (Figure 2-(b)) or replacing each $4\times C_1$ by $1\times C_3$ (Figure 1-(c)).

The $STBN(3,0)$ shown in Figure 1 can use $4\times C_1$, $2\times C_2$, or $1\times C_3$ for each bundle of first stage, $2\times C_1$ or $1\times C_2$ for each bundle of second stage, and only $1\times C_1$ in third stage. The $STBN(3,1)$ shown in Figure 2 can use

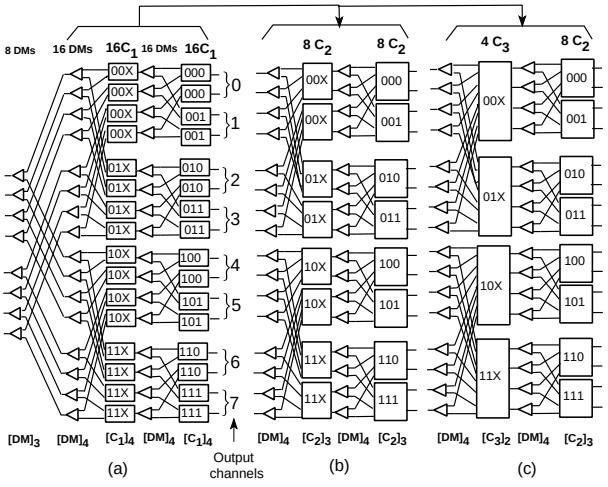

Figure 2. STBN(3,1) with different blocking degrees

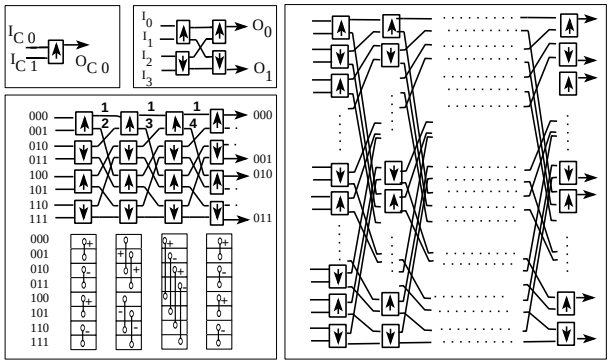

Figure 3. Concentrators 2×1, 4×2, and 8×4.

$4\times C_1$, $2\times C_2$, or $1\times C_3$ for each bundle of second stage, $2\times C_1$ or $1\times C_2$ for each bundle of third stage.

Reducing blocking can be made by using larger concentrators. For example the CLP can be reduced for a set of 8 inputs if instead of using four 2×1 concentrators we use two 4×2 concentrators or one 8×4 concentrator. A STBN in which the degree of expansion of each link is d (1:2^d STBN) is said to have 2^d-dilation. Figure 2 shows an 8-input 1:2 STBN that has 3 stages and 8 output channels allowing up to 2 cells to be simultaneously transmitted over distinct links of a given channel. Each output channel has 2 ports (4-dilation).

3. THE CONCENTRATOR

In general our concentrator $C(n,n-1)$ transmits at most 2^{n-1} cells among its most prior 2^n input cells and will be denoted by C_n. The filtering function within the concentrator is based on cell priority. Ideally, $C(n,l)$ allocates up to its 2^{n-1} output ports to the most prior 2^{n-1} cells among the 2^n cells present at its inputs.

4. COMPLEXITY ANALYSIS

Table 1 presents the $STBN(n,d)$ input-output delay and total number of DMs and sorters. The delay and total number of DMs denoted by τ_{DM} and N_{DM}. τ_{FS} denotes the

Table 1. Delay and cost in $STBN(n,d)$.

STBN	In-out delay	Total Cost
(n,d)	$\tau_{DM} = nt_{DM}$	$N_{DM} = 2^n(2^d(n-d+1)-1)$
(n,d)	$\tau_{FS} = \tau_S(2^n - 2^d)$	$N_{FS} = 2^{n+d}(2^n - 2^d)$
(n,d,k^+)	$\tau_{S+} = t_S((n-k+2)2^{k-1} -2^d)$	$N_{S+} = (n-k)2^{n+d+k-1} + 2^{n+2d}(2^{k-d}-1)$
(n,d,k^-)	$\tau_{S-} = t_S(2^k + n - d - k - 1)$	$N_{S-} = 2^{n+d} \times (2^k + n - d - k - 1)$

Table 2. Throughput of sorter vs load and traffic priority.

	INPUTS		OUTPUTS	
	load pr=0/1	Load pr=1	Throughput pr=0/1	Throughput pr=1
Upper	q_0	p_0	$q_0^+ = q_0 + q_1 - q_0q_1$	$p_0^+ = (q_0p_0 + q_1p_1 - q_0q_1p_0p_1)/q_0^+$
Lower	q_1	p_1	$q_1^+ = q_0q_1$	$p_1^+ = (q_0q_1p_0p_1)/q_1^+$

total delay due to sorter stages and N_{FS} denotes the total number of sorter stages. To limit the cost we may set an upper bound on the concentrator bandwidth by using C_k in stage $1 \le s \le n-d$ whenever $k \le n-s+1$, as described in the structural connectivity of $STBN(n,d,k)$. We have two options for the distribution of C_ks.

$STBN(n,d,k^+)$ uses C_k in stages 1 to $n-k+1$, C_{k-1} in stage $n-k+2$, ..., C_{d+1} in stage $n-d$. In the concentration phase, τ_{S+} denotes the total delay due to sorter stages and N_{S+} denotes the total number of sorter stages. $STBN(n,d,k^-)$) uses C_k in stage 1, C_{k-1} in stage 2, ..., C_1 in stages k to $n-d$. In the concentration phase, τ_{S-} denotes the total delay due to sorter stages and N_{S-} denotes the total number of sorter stages.

5. ANALYTICAL MODEL

The passthrough probability of the sorter is given in Table 2, where q_0, p_0, q_1, p_0 are the load on upper sorter input, load of higher-priority traffic on upper input, load on lower input, and load of higher-priority traffic on lower input, respectively. Output variable are denoted with $+$. The throughput of the dilation phase is $T_{dilation} = q2^{-d}$. The throughput of the concentration phase ($T_{concentration}$) can be recursively obtained by computing the throughput of each stage as function of input load and iterating along $n-d$ stages.

6. EVALUATION AND COMPARISON

In this section we study CLP and service rate under uniform traffic of STBN, at full load, versus switch size, concentrator bandwidth, and dilation degrees.

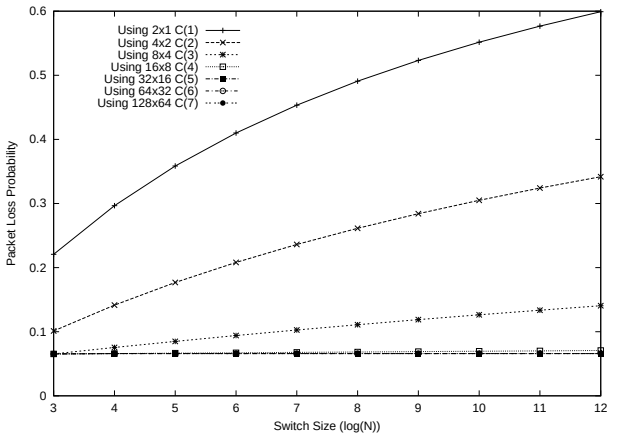

Figure 4. CLP of PTBN under full load uniform traffic versus switch and concentrator sizes for $d = 1$

Figures 4 present the CLP of STBN for dilation $d = 1$ $(1:2)$ versus switch size (8×8 to 4096×4096) and different concentrator bandwidth C_k $(1 \le k \le 7)$, i.e. concentrator size ranging from a 2×1 to 128×64. The CLP is moderately dependent on switch size when using moderate concentrator bandwidth (C_k where $k = 1$). The same effect exists for all banyan networks due to internal conflicts. However, most internal conflicts in the STBN seem to be avoided when using higher concentrator bandwidth (C_k where $k \ge 3$). For low dilation degrees, increasing the concentrator bandwidth moderately decreases the CLP. There is little benefit from using 0-dilation and a concentrator C_k with $k \le 4$, i.e. a 16×8 concentrator. For moderately higher dilation ($d = 1$ or 2), increasing the concentrator bandwidth significantly decreases the CLP. For example an STBN with 4096-input/output has a CLP of 0.4 when it using C_1 and a CLP of 10^{-4} when using C_7.

Increasing the dilation degree alone produces a moderate decrease in CLP. However, a small increase in the concentrator bandwidth (C_k where $k \le 3$) combined with a moderate dilation degree (for example $d = 1$ of 2) produces significant decrease in CLP by up to 10^{-3} folds the standard banyan, i.e. an STBN(n,0,1) for which all concentrators are C_1.

The STBN becomes very selective in bandwidth allocation when its passthrough is relatively low. Due to the use of concentrators, the STBN gives higher priority in allocating its outputs to higher-priority traffic. However, when the passthrough of STBN is high (low CLP) nearly all the input cells are switched by the STBN, thus making the load of higher-priority traffic at input very close to that at STBN output.

Figures 5 and 6 presents the *Service Rate (SR)* which is the ratio the number of cells switched per unit of propagation delay. SR grows linearly with the switch size whenever the concentrator size is valid for STBN (C_k with $1 \le k \le n$). Notice SR crossover in performance caused by tradeoff between throughput and delay for $n \ge 10$ using C_1 and C_2.

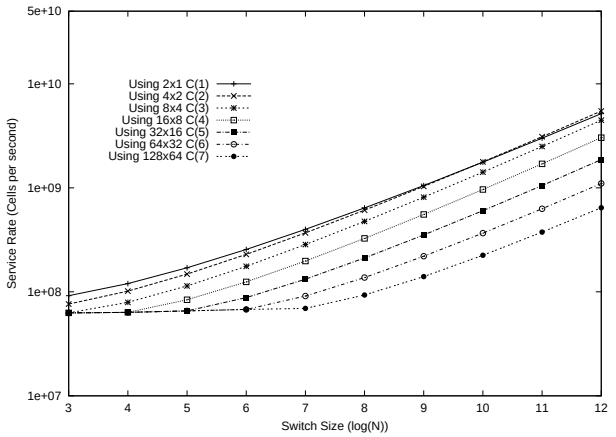

Figure 5. Service Rate ($d = 0$) versus switch size and C_k

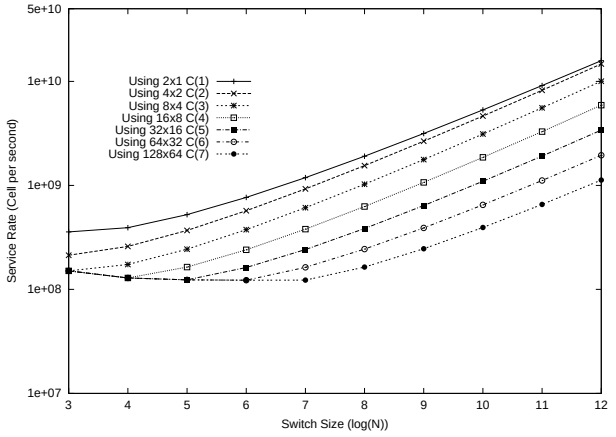

Figure 6. Service Rate ($d = 2$) versus switch size and C_k

In [4] non-blocking requires complex hardware and excessive processing time to detect and extract conflicting cells and re-assign them to new inputs. Similarly in [3] the complexity is in dynamic re-routing from active ports to idle ports. In [5] excessive complexity is used 3D switch where multiple crossbars are connected in tandem at all crosspoints. While the STBN provides scalable performance using traditional banyan connectivity, it is expected to achieve a switching rate of Giga-cells/second (Fig. 5 and 6) using simple self-routing. Moreover the STBN provides a scalable solution for many high-speed switches like the *Pipeline Banyan* [6] which employ the standard banyan.

7. ACKNOWLDGEMENT

Thanks to Students H. Tahhan and M. Senaryo for their programming simulation. Thanks to the College of Computer Science and Engineering and KFUPM for computing support.

8. CONCLUSION

The PTBS switch is based on successive (1) routing that expands the paths using multiplexers, and (2) selecting path allocation using concentrators. Evaluation shows that

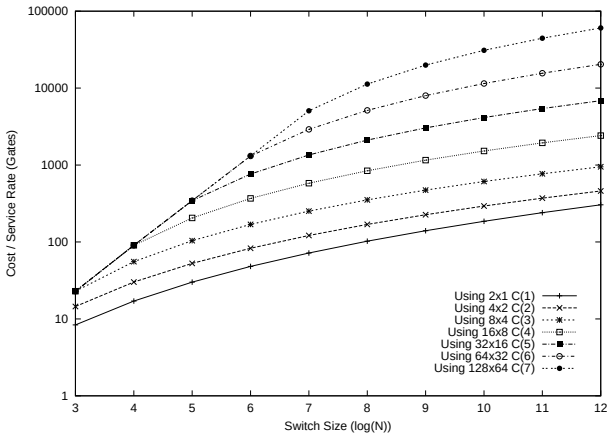

Figure 7. Cost in gates per 1 Gcps ($d = 0$) vs. switch size and C_k

a small increase in the concentrator bandwidth combined with a moderate dilation degree produce a significant decrease in CLP by up to 10^{-3} fold the standard banyan. This shows that the STBN provides an effective tool to the scalability of banyan networks. Due to the use of higher order concentrators, the STBN becomes very selective in bandwidth allocation when its passthrough is relatively low giving higher priority in allocating its outputs to higher-priority traffic. This provides some Q.o.S guarantees for selectively switching real-time traffic while providing a scalable switching architecture.

9. REFERENCES

[1] Ronald J. Vetter. ATM concepts, architectures and protocols. *Communications of the ACM*, 38(2):31–38, Feb 1995.

[2] M. Al-Mouhamed and M. Kaleemueddin. Evaluation of Pipelined Switch Architecture for ATM Networks. *IEEE/ACM Trans. on Networking*, No 5, Vol 7:724–740, Oct 1999.

[3] A.M. Lele and S.K. Nandy. Architecture of a reconfigurable low power gigabit atm switch. *14th International Conference on VLSI Design*, pages 242 –247, 2001.

[4] Joo-young Lee and Jae-il Jung. Design of non-blocking permutation generator. *IEEE International Conference on Communications*, 4:2090–2094, 2002.

[5] E. Oki and N. Yamanaka. Tandem-crosspoint atm switch with input and output buffers. *IEEE Communications Letters*, 2:189–191, 1998.

[6] P. C. Wong and M. S. Yeung. Design and analysis of a novel fast packet switch–Pipeline Banyan. *IEEE/ACM Transactions on Networking*, 3(1):63–69, Feb. 1995.