Evaluation of Pipelined Switch Architecture for ATM Networks

M. Al-Mouhamed, M. Kaleemuddin, H. Yousef [*]

Abstract

In this paper, we present an investigation of pipelined switch architectures employing a family of *Dilated Banyans.* Pipelined banyan [1] was proposed earlier for increasing throughput and reducing switching delay in banyan-based ATM switches. Our objective is to find a switch architecture that maximizes the *service rate* which is one structural feature defined by the ratio of throughput to switching delay. Achieving an acceptable *cell loss probability* (CLP) in pipelined banyan requires the use of a number of reservation cycles which determine overall switching delay. This allows finding an upper bound on the service rate for a given design methodology for which an acceptable CLP is guaranteed. For higher cell arrival rates, we present a family of dilated banyans for which the throughput and propagation delays can be scaled up with additional hardware. There are two extremes for dilated banyans in which a channel may have: (1) few high-bandwidth links, or (2) many low-bandwidth links. Therefore, the need to find a pipelined dilated banyan that maximizes the service rate. For this we carry out complexity analysis and simulation of few pipelined dilated banyans which we subject to *uniform traffic* and *ATM traffics* and study performance under variation in load, buffer size, and number of data planes. An ON-OFF model is used for the generation of ATM traffics. Compared to pipelined banyan, pipelined dilated banyans can provide up to 4 times the service rate without dramatically increasing the switching delays. This provides insight on banyan architectures that are most suitable to miximize the service rate in pipelined switch architectures.

1 Introduction

Asynchronous Transfer Mode (ATM) is the transport mode of Broadband-ISDN (integrated services digital network) [2, 3, 4]. ATM is a cell switch technology. At the source end system, the traffic stream is split into small fragments of 48 bytes each. A fragment together with a five byte header make up an ATM protocol data unit called a *cell.* The 5 byte header contains ATM protocol information required to deliver the cell to the destination end system across an ATM network. ATM prides itself of being the base of future

[*] The authors are with the Department of Computer Engineering, College of Computer Science and Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia.

ubiquitous computer networks. It is designed to operate over high speed and is limited only by the technological barriers of transmitting hardware and links.

The connection oriented nature of ATM, together with the use of statistical multiplexing and fixed size small cells allow ATM to adequately support real-time and non-real-time communication. ATM uses the concept of virtual connections between end-stations [5, 6]. Two types of connections are possible: permanent and switched. A permanent virtual circuit (PVC) is a connection that is manually established and manually released. End-stations do not have the ability to do that dynamically. A switched virtual circuit (SVC) is a connection that is dynamically established and released via signaling as required. Virtual connections are identified by their virtual paths and virtual channels. A virtual path is a logical construction of a group of virtual channels. The cell header combines both a virtual path identifier (VPI) and a virtual channel identifier (VCI). These identifiers will guide the cell through the ATM network

Besides its connection oriented nature, ATM has a number of unique and desirable features: (1) It provides the speed on a per-source basis, which means that each source can have its own high speed; (2) ATM is scaleable, since it can provide higher speeds to applications that require it; and (3) flexible, since we can mix speeds within a network according to the user requirements. The speed mixing is handled by the ATM equipment automatically.

ATM connections are established with negotiated quality of service (QoS) requirements, thus enabling real-time communication service such as videophony and video conferencing. Quality of Service is a unique feature of ATM. Depending on the application requirements, workstations can request specific QoS parameters for the connections they are going to setup. Five service categories are supported by ATM systems: (1) Constant Bit Rate (CBR), (2) Variable bit rate-real time (VBR-rt), (3) Variable Bit Rate-non real time (VBR-nrt), (4) Available bit rate (ABR), (5) and Unspecified bit rate (UBR). The source category is based on the declared QoS parameters, namely the Peak Cell Rate, the Sustainable Cell Rate, and Maximum Burst Size.

ATM technology is gaining ground, as more systems are getting deployed, and many of the complex issues getting resolved and standardized. One of the difficult problems addressed by the industrial and research communities is the engineering of ATM switches capable of accommodating a variety of traffic sources with conflicting quality of service requirements, and which can scale up in performance to rising bandwidth needs.

Banyan networks, a class of *multistage interconnection networks* (MINs), have several desirable features such as space division, self routing, low hardware complexity, and regular structure which make them suitable for VLSI implementation. Unfortunately, their throughput is far from being acceptable due to *internal* and *external* blocking. Several strategies have been suggested in order to overcome these problems. One is to use internal buffering [7] but this increases the switch complexity which result in large hardware overhead as well as *Head of Line* (HOL) queuing delays. Internal blocking can be avoided totally if we use sorting network in front of the routing network. This has been done in the Batcher-Banyan network [8]. But the problem here is the complexity [6] of the sorting network. Moreover output conflicts are still there. An early strategy [9] to increase throughput is to distribute the incoming traffic over *parallel Banyans*, to decrease the load on each banyan, so that the routed cells are forwarded to the corresponding output buffers. Here the throughput increases slowly with the number of planes because internal conflicts can still occur in each banyan.

In the *Tandem Banyan Switching Fabric* (TBSF) [10] the cells are issued to banyans arranged in series. One conflicting cell is misrouted in current banyan and re-issued to next banyan. Thus by properly adjusting the number of banyans the *cell loss probability* (CLP) can be made as low as needed. The cost is the relatively large number of banyans and implied propagation delays.

Multi-parallel banyans use vertical connections to shorten propagation delays. This is case of *Piled Banyan Switching Fabric* (PBSF) [11] and *Prallel-Tree Switching Fabric* (PTBSF) [12] which have no input buffering. In case of cell conflict vertical cell forwarding significantly reduces the propagation delay. However, achieving low CLP (bounded for the PBSF) requires complex hardware and large number of connections.

The *Pipelined Banyan* (PB) [1] uses one single control plane for path reservation and a number of data planes for payload transfer. Each time slot consists of a number of reservation slots. In a reservation slot, an input buffer may be notifyied to re-submit its cell header in next slot if its cell (header) cannot continue its self-route because of conflicts. A successfully self-routed header makes reservation of a path on a free data plane to transfer its payload. Since cell header in ATM is much shorter than payload, therefore, multiple reservation slots can be done during one payload transfer slot which enables pipelined operations. Thus, PB has low switching delay and relatively high throughput.

In this paper, we present an investigation of pipelined switch architectures employing *Dilated Banyans* (DBs) with the objective of finding an architecture with the highest possible service rate. The primary DB was originally proposed [13] for the *Burroughs FMP* multiprocessor and later studied [14] as an unbuffered shuffle-exchange network and in packet switching in [15]. We show how the architecture of DBs can be made scalable with respect to throughput and propagation delays. We study pipelined switches employing few DBs for which we evaluate the: 1) switching delays, 2) number of needed data planes to guarantee some CLP level, and 3) overall hardware complexity. This will allow us to find some pipelined schemes employing specifically designed DBs that can provide high throughput with reasonable switching delay.

The organization of this paper is as follows. Section 2 presents some background on related switch architectures. Section 3 presents the topology and complexity analysis the proposed dilation banyan within the pipelined scheme. Section 4 presents evaluation of the proposed switch architecture under uniform traffic. In section 5 we present performance of proposed switch under some ATM traffic mixes. In Section 6 we conclude about this work.

2 Background

One fundamental probem to minimize CLP in ATM switches is to find an efficient method for partitioning the set of HOL cells into subsets so that the cells within each subset are free of internal and external conflicts. Each subset of cells can then be switched out without conflicts by using a separate banyan. Unfortunately, there is no efficient method to partition the cells and routing the subsets in parallel. A number of proposals have been made to provide partial solutions. The idea is to issue all HOL cells to one banyan, perform self-routing, retrieve cells which reach their destinations, and re-issue all unsuccessful cells to the banyan, and so on. We call this approach *Iterative Conflict Resolution* (ICR) in which a cell loss occurs in: 1) last banyan in the case of no input buffering, or 2) input buffers in the case of full buffers.

The ICR strategy can be used to provide arbitrary low CLP by selecting an appropriate number of iterations for a given switch size. Example of switch architectures employing the ICR method is the *Tandem Banyan Switching Fabric* (TBSF) [10] in which cells are applied to banyans arranged in series. When a conflict occurs within some banyan, one of the cells is routed correctly while the other is misrouted and marked as such. At the banyan output, those cells that arrive to each output ports without being misrouted are forwarded to coresponding output buffers. The misrouted cells are applied to the next banyan in series. Cells reaching incorrect output of last banyan are lost. This architecture can achieve arbitrary low CLP at the cost of relatively long switching delay.

Multi-parallel banyans use vertical connections among banyans to reduce sequential propagation delays. The *Piled Banyan Switching Fabric* (PBSF) [11] and *Parallel-Tree Switching Fabric* (PTBSF) [12] are two examples. If two cells conflict within an SE one cell is routed to the correct output and the other is routed to the corresponding SE of a banyan located in the next lower level. Each SE has two horizontal inputs and outputs as well as two vertical inputs and outputs. By avoiding misrouting of cells, vertical links shorten the path delay. Cell loss can occur in: (1) SEs of arbitrary banyan of PBSF, and (2) in SEs of lowest banyan of PTBSF. Hence, the throughput of the piled banyan saturate at 98% under full load, while the PTBSF can scaled up to achieve arbitrary low CLP. Both switches have short porpogation delay but they use relatively large amount of hardware and interconnections.

A recent switch that addresses the above issues is the *Pipelined Banyan* (PB) [1]. It consists of one single control plane and a number of data planes. In a reservation slot, headers of all HOL cells are self-routed to their destinations within the control plane. In the case of conflict between two headers one of them is dropped and a back-pressure mechanism is used to notify the corresponding input buffer to re-submit its cell header in next slot. Headers that successfully reach their destinations make reservation of paths on a data plane to transfer their corresponding payloads from input buffers to destination output buffers. Since cell header in ATM is much shorter than payload, therefore, multiple reservations can be done during payload transmission time. Self-routing of cell headers without payload contributes in shortening the reservation time. Thus the reservation time and cell transmission time can overlap which enables pipelined operations. It is clear that the presence of separate control and data planes significantly contributes in reducing switching delay and pipelining helps in improving overall switch throughput.

A banyan used in pipelined scheme is characterized by its throughput, delay in control plane, and delay in data plane. The number of reservation slots in each time slot is dictated by the need to achieve some guaranteed CLP. This determines the time slot if one can estimate the delays in control and data planes. The ratio of switch throughput to duration of time slot is the *service rate* of the pipelined switch which is one structural feature of the banyan used. Thus there is an upper bound on the service rate for a given banyan architecture. The pipelined switch can still guarantee some CLP if overall cell arrival rate is below the service rate of the switch. With inceasing cell arrival rate there is need for a switch that can provide higher throughput without dramatically increasing propagation delays. Our objective is to investigate a class of dilated banyan architectures that are capable of producing very high throughput by adjusting channel bandwidth. There are two extremes in which a channel has: (1) few high-bandwidth links, or (2) many low-bandwidth links. Between the two extremes lies a class of DBs that can be engineered to produce a wide variety of service rates when used in the pipelined scheme.

In the next section we present a class of DBs, investigate their basic building blocks, analyse their hardware complexity, and build a delay model for their modular components. The latter will be used to estimate propagation delays in control and data planes.

3 Pipelined dilated banyans

Our objective is to finding a pipelined switch architecture having high service rate. For this we study in this section a class of DBs to be used as banyan planes in the pipelined scheme. We provides the basis for general evaluation of hardware complexity and propagation delays of arbitrary banyans made of switching elements having 2 input and 2 output channels where each input or output channel has power-of-2 number of ports.

In the next sub-section, we present a class of DBs for which the building blocks are: (1) 2×2 *binary sorter*, and (2) 1×2 *demultiplexer*. Next we study the complexity of DBs through evaluation of the number of sorters and demultiplexers, interconnection links, and propagation delays. Using a design approach for the sorters and demultiplexers, we also find the complexity of needed hardware in terms of number of gates and express propagation delay as function of gate delay.

3.1 The dilated switch

The architecture of DBs is based on expanding the internal channel bandwidth (multiple ports) to reduce the CLP. A DB in which the degree of expansion of each link is d (1:2^d DB) is denoted by 2^d-dilation. Figure 1 shows an 8-input 1:4 DB that has 3 stages and 8 output channels with up to 4 cells can be simultaneously transmitted over distinct links of a given channel. Each output channel has 4 ports (4-dilation). The dashed lines show one path through the binary expansion. The first two stages expand the input ports in the form of a binary tree by using demultiplexers. The third stage is a routing phase for which the basic switch is called (D-SW) switch which has 2 input channels and two output channels. In the case shown in Figure 1 each channel of the D-SW switch has 4 ports. There is no cell loss in the first two stages but cell loss can occur only in any D-SW of the routing phase.

The general architecture of an n inputs 1:2^d DB is shown on Figure 2, where d always satisfies $0 \leq d \leq n$. It has two phases: 1) *expansion phase* (non-blocking), and 2) *routing phase* (blocking). In the expansion phase the internal links are doubled at each of the d stages until reaching a 2^d-dilation at the dth stage. The kth stage of the expanssion

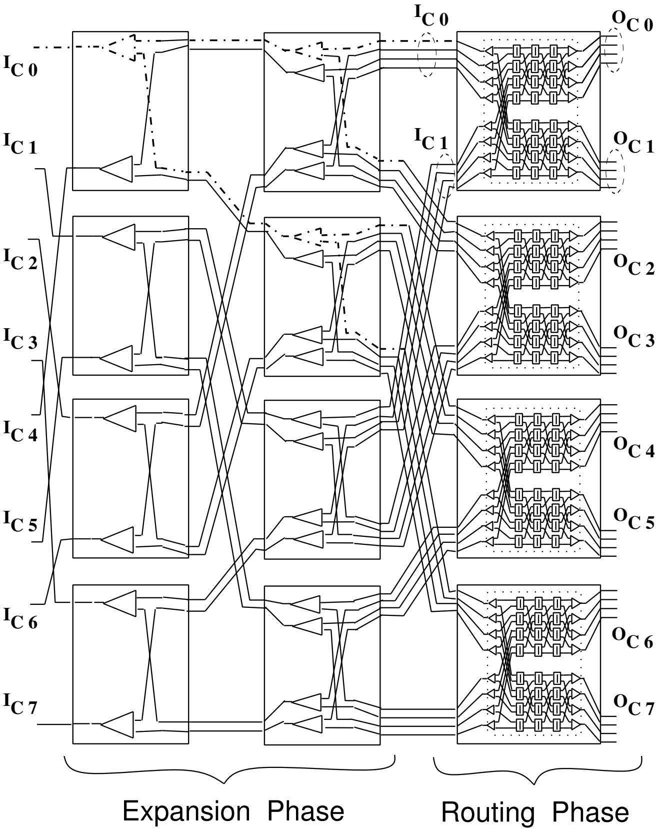


Figure 1: 1:4 Dilated Banyan with 8 inputs (each output channel has 4 ports)

consists of $2^n \times 2^k$ demultiplexers (DM) which are 1×2. Each DM routes an incoming cell to its upper or lower outputs depending on the cells destination bit.

The routing phase consists of $n - d$ stages, each contains 2^{n-1} switches called *D-SW* switches as shown on Figure 2. Each *D-SW* switch has two input channels (I_{C0} and I_{C1}) and two output channels (O_{C0} and O_{C1}) and each of these channels has 2^d ports. Input and output channels are shown on Figure 1 for $n = 3$ and $d = 2$. At most 2^d cells with identical output can be simultaeously transmitted (one cell per port) without conflicts from input to output. A DB switch becomes non-blocking if the dilation degree is equal to number of stages, i.e. there is no routing phase. In this case the hardware requirements are maximal due to its n-level 2^n binary trees. On the other hand, a DB without an expansion phase is a simple banyan switch (1:1 or 1-dilation).

The *D-SW* is a 2 input and output channels where each channel has 2^d ports. Internally, the *D-SW* switch consists of one stage of demultiplxers (DM) and two binary concentrators called *Upper Concentrator* (UC) and *Lower Concentrator* (LC). Each of the UC and LC may at most have 2^d stages as will be shown latter. The maximum throughput of the *D-SW* switch (see Appendix) corresponds to 2^d stages in each of UC and LC. One may consider *D-SW* switches in which UC and LC are k-stage networks which we denote by $D\text{-}SW(d, k)$.

Simple banyans use $D\text{-}SW(0, 1)$, shown in Figure 3-(a), as a 2×2 switching element

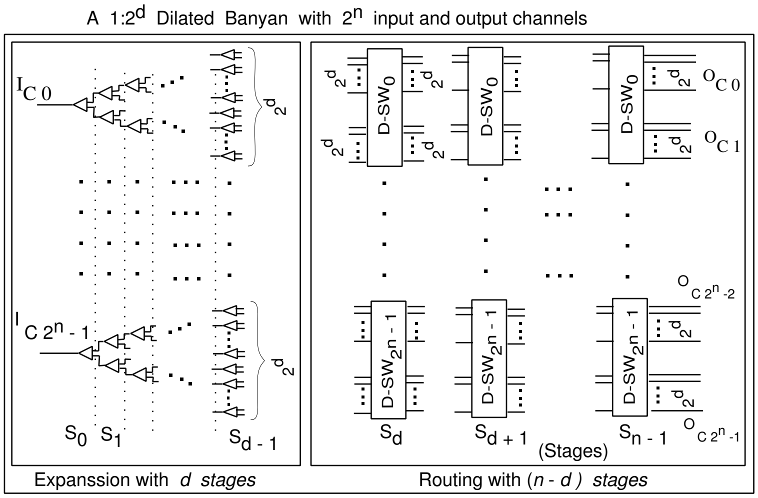


Figure 2: General architecture of the expansion and routing phases

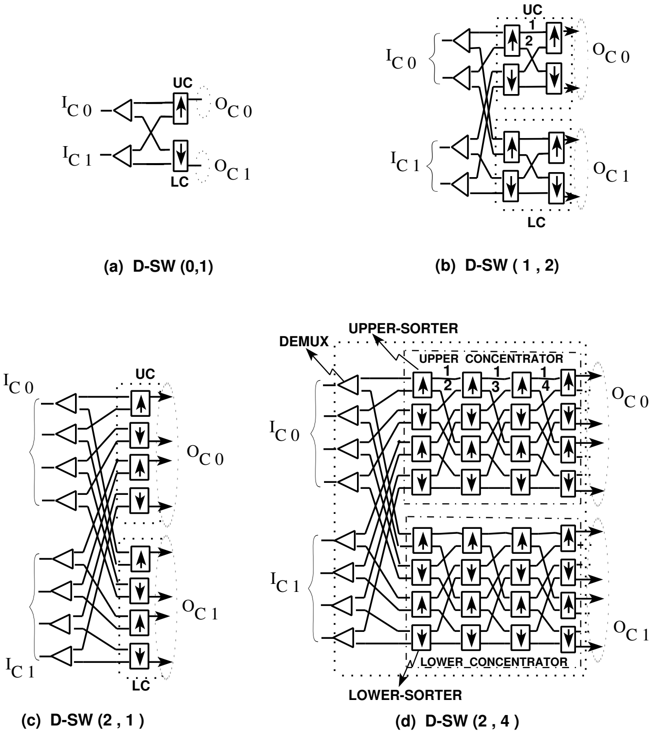
(SE). A 2-DB uses $D\text{-}SW(1,2)$, shown in Figure 3-(b), as basic SE. Figures 3-(c) and (d) show two $D\text{-}SW(2,k)$ for which $k = 1$ and $k = 4$, respectively. In $D\text{-}SW(2,1)$ a loss may occur if there are two conflicting cells on two successive input ports of any given channel. In $D\text{-}SW(2,4)$ a loss may occur if there are more than four conflicting cells on both channels. Between $D\text{-}SW(2,1)$ and $D\text{-}SW(2,4)$ lies two other $D\text{-}SW$s ($k = 2$ and $k = 3$) with intermediate bandwidth. For example, in $D\text{-}SW(2,2)$ a loss may occur if there are three conflicting cells on four input ports of any given channel.

For maximum throughput a 4-DB may use $D\text{-}SW(2,4)$ as the basic SE, but $D\text{-}SW(2,1)$ or $D\text{-}SW(2,2)$ or $D\text{-}SW(2,3)$ could also be used to reduce cost. An n-stage, 1:2^d DB in which the routing phase is made of $D\text{-}SW(d,k)$ is denoted by $DB(n,d,k)$, where $k \leq 2^d$. For example, the DB shown in Figure 1 is $DB(3,2,4)$.

The UC and LC have identical architecture and each has 2^{d+1} input ports and 2^d output ports. Depending on cell destination bit (x), the DM stage routes incomming cells (up to 2^{d+1} cells) to one of the 2^d input ports of UC (for cells with $x = 0$) or to one of the 2^d input ports of LC (for cells with $x = 1$). Each of UC and LC sorts incomming cells (at most 2^{d+1} cells) based on cell priority bit and exit at most 2^d cells.

A $D\text{-}SW(n,d,2^d)$ outputs at most 2^d cells among the most prior cells. In this case, in each of UC and LC the lower priority cells in excess of 2^d are internally discarded. In other terms, each concentrator takes at most 2^{d+1} inputs cells, select at most 2^d cells among the most prior regardeless of their source input port, and forward the selected cells to its 2^d output ports. Therefore, cell loss can occur within $D\text{-}SW$ only if there are more than 2^d cells at the input of the $D\text{-}SW$. The routing of the $D\text{-}SW$ switch ensures the highest possible throughput but other routing disciplines are also possible as in the case of $D\text{-}SW(n,d,k)$ where $k < 2^d$.

The UC and LC are made of stages of 2×2 binary sorters (up-sorters and lower-sorters) interleaved with a perfect-shuffle permutation. If there is two input cells to up-sorter (lower-sorter), then the cell with highest priority exits at the upper (lower) output and the other cell exits at lower (upper) output. A single input cell is sent to



(a) D-SW (0,1)

(b) D-SW (1 , 2)

(c) D-SW (2 , 1)

(d) D-SW (2 , 4)

Figure 3: The family of D-SW switches: D-$SW(0,1)$ (a), D-$SW(1,2)$ (b), D-$SW(2,1)$ (c), and D-$SW(2,4)$ (d)

upper output regardless of its priority. The up-sorter and down-sorter have symmetric functions.

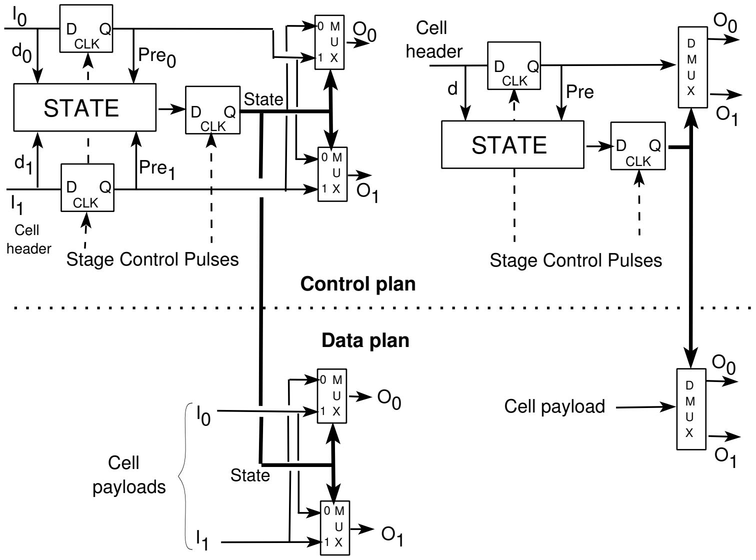
3.2 Complexity analysis

In this section we present the hardware requirements, number of interconnection links, and time delay along a path for the class of $DB(n,d,k)$. These parameters are useful for evaluating the service rate as well as for comparing to other switches. The detail of complexity analysis is presented in the Appendix.

In a 1:2^d DB, there must be 2^d sorter stages in each concentrator if no cell loss should occur within the concentrator (last stage) when at most 2^d cells are present on the 2^{d+1} input ports of concentrator. In other words, the throughput of a D-$SW(d,k)$ scales up with increasing the number of sorter stages up to some value $k = 2^d$.

In a 1:2^d DB, there are $DB_{Dmux}(n,d,k) = 2^{n+d}(n-d+1) - 2^n$ Demultiplexers and $DB_{sorter}(n,d,k) = k(n-d)2^{n+d}$ Sorters. A $DB(n,d,k)$ made of demultiplexers and sorters has $DB_{Link}(n,d,k) = 2^{n+d}[(n-d)(2k+1)+2] - 2^n$ interconnection links.

In a pipelined DB, cell headers make path reservation in the control plane prior to performing payload transfer in data plane. This means we need to distinguish between switching delays in control plane and data plane. The switching Delay along a path from input to output of a 1:2^d n-stage $DB(n,d,k)$ is $\tau_{control}(n,d,k) = [3n + 10k(n-d)]\tau_{gate}$ in



a - A sorter in control plan (up) and in data plan (down).

b - A DM in control plan (up) and in data plan (down).

Figure 4: Sorter (a) and DM (b) switches in control and data plans

Hardware Resources	Simple Banyan $BD(n,0,1)$	Partially-dilated banyan $BD(n,d,1)$	Fully-Dilated banyan $BD(n,d,2^d)$
$DB_{Sorters}$	$n2^n$	$(n-d)2^{n+d}$	$(n-d)2^{n+2d}$
DB_{Dmux}	$n2^n$	$(n-d+1)2^{n+d}-2^n$	$(n-d+1)2^{n+d}-2^n$
DB_{Link}	$(3n+1)2^n$	$[3(n-d)+2]2^{n+d}-2^n$	$[(n-d)(2^{d+1}+1)+2]2^{n+d}-2^n$
$\tau_{control}$	$23n\tau_{gate}$	$(23n-12d)\tau_{gate}$	$[11n+12(n-d)2^d]\tau_{gate}$
τ_{data}	$3n\tau_{gate}$	$(3n-2d)\tau_{gate}$	$[n+2(n-d)2^d]\tau_{gate}$

Table 1: Hardware resources required in dilated banyans

the control plane and $\tau_{data}(n,d,k) = [n+2k(n-d)]\tau_{gate}$ in the data plan, where τ_{gate} is one gate delay time.

We now evaluate the number of gates required for the sorter and DM swithces in the control as well as the number of gates for their corresponding hardware in the data plan. Figure 4-a shows the sorter hardware in both control and data plans. The state of the sorter depends on two cell presence bits (pre_0 and pre_1) and two priority bits (pr_0 and pr_1) for its two input cells. For this the sorter needs 3 D-latches which require 12 dual-ported gates. The state requires 3 dual-ported gates. Also two 2 × 1 MUXs are needed which requires 6 additional gates. The total number of gates for the sorter in control plane is $N_{Sorter}^{control} = 21$ gates. In data plane we need two 2 × 1 multiplexers which requires $N_{Sorter}^{data} = 6$ gates.

The demultiplexer DM requires two D-latches (head bit and state bit), state logic, and one 2 × 1 DMUX. The state is function of cell's presence and destination. Using dual-ported gates, 8 gates are needed for the D-latches, 1 gate for the state, and 2 gates for the DMUX. The total number of gates for the DM in control plane is $N_{DM}^{control} = 11$

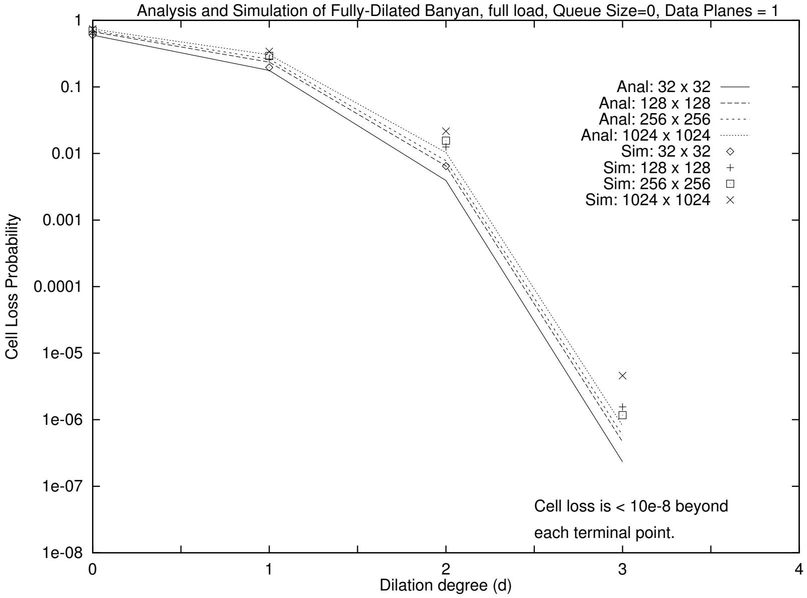


Figure 5: CLP of fully-dilated banyans versus dilation degree

gates. In data plane we need one 1 × 2 MUX which requires $N_{DM}^{data} = 2$ gates.

We evaluated hardware complexity and delays for a number of useful DBs which are *simple banyan* ($DB(n, 0, 1)$), *partially-dilated banyan* ($DB(n, d, 1)$), and *fully-dilated banyan* ($DB(n, d, 2^d)$). The concentrator of each *D-SW* consists of: (1) only one stage ($k = 1$) of sorters for the partially-dilated banyan, and (2) 2^d stages of sorters for the fully-dilated banyan. Table 1 summarises the hardware complexity and delays for the above DBs.

In this Section we presented the architecture of DBs together with a design model to evaluate the propagation delays. In the next Section we carry out simulation of pipelined DBs under uniform traffic and find the number of needed reservation cycles to achieve some CLP. This will enable evaluation of service rate and comparison.

4 Evaluation under uniform traffic

In this section we study the CLP and cell delay of pipelined switch architectures under uniform traffic pattern. We assume time slotted synchronized operations for which the slot time is greater than or equal to the switch processing time. Cells must be synchronized and aligned with the local slot boundaries before being routed by the switch. The workload assumptions are as follows. All switch inputs are identical and independent. In every time slot, each input buffer has a probability p of receiving a new cell and $1 - p$ of receiving no cells. We also assume that the activity level at the various inputs are independent in time as well as in space. The cell destinations are uniformly distributed over all the switch outputs.

In pipelined switches the cells are generated in the begining of each time slot which consists of a number of reservation slots. HOL cells are submitted to control plane in each reservation slot. In unbuffered switches only cells generated during the current time slot

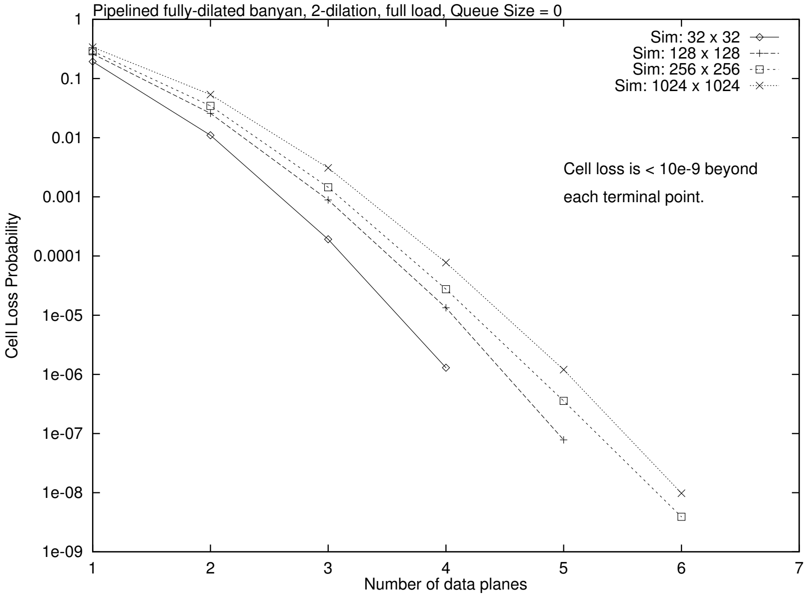


Figure 6: CLP of unbuffered pipelined fully-dilated banyans

are submitted to the control plane for each reservation slot until they succede in reserving a path or being considered as lost. In buffered switches the same process occurs with the difference that an HOL cell runs for each reservation slot and remains as HOL if it fails in making a reservation. Therefore, cell loss can occur in a buffered switch only if a cell is generated for a given input buffer and that buffer is full.

In the next sub-sections we study performance of pipelined switches employing: 1) fully-dilated banyans, 2) partially-dilated banyans, and 3) simple banyans.

4.1 Performance of pipelined fully-dilated banyans

The partially-dilated banyan $DB(n, d, 1)$ has the least throughput and least hadware requirements among all DBs with n-stages and 2^d-dilation. At the other extreme we have the *fully-dilated banyan* $DB(n, d, 2^d)$ for which there are 2^d-sorter stages in each *D-SW* switch of the routing phase. The throughput of $DB(n, d, 2^d)$ is the highest among the family of DBs. In this section we evaluate performance of *Pipelined Fully-Dilated Banyan* (PFDB) under uniform traffic. This means we use one $DB(n, d, 2^d)$ in control plane and in each data plane of the pipelined switch.

Figure 5 shows the analytical and simulation performance of single $DB(n, d, 2^d)$ for some values of dilation degree d. See the Appendix for a derivation of the analytical model of $DB(n, d, 2^d)$. Here a dilation degree of 3 (8-dilated) is sufficient to achieve a CLP around 10^{-6} at full load but at high cost of the hardware compared to that of simple banyan. This is discussed later in the hardware analysis section. Note that fully-dilated banyans exhibits nearly the same CLP regardless of the switch size.

The analytical results were more optimistic than simultation results because the analytical model assumes uniform traffic at all the stages. In reality the uniform traffic gets corrupted by the deterministic stage routing which increases from one stage to the

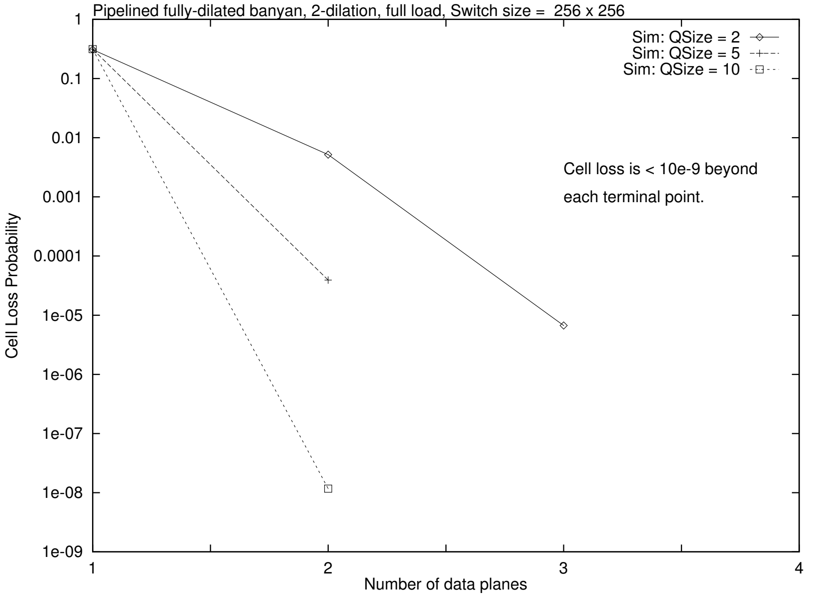


Figure 7: CLP of buffered pipelined fully-dilated banyans

next. In other terms, the traffic uniformity decreases with increasing stage number. We validated the above interpretations by uniformly re-generating the cell destination after each stage of the simulation which gave CLPs that were very close to those obtained from the analysis. Therefore, the simulation results are more representative of real switch performance than the analytical results. We also evaluated the analytical model of pipelined DBs by using our analytical model of $DB(n, d, 2^d)$ and the *discrete-time probability state transition* used in [1] to model the number of backlogged cells in input queues.

Buffering of single plane DB did not produce significant drop in CLP except when the dilation degree was 3 or above, i.e. when CLP is very low. For the pipelined scheme we used fully-dilated banyan with dilation degree 1 (2-dilation) and observed good results. As shown on Figure 6 an unbuffered PFDB requires only 5 or 6 data planes as compared to unbuffered PB. The use of input buffering significantly reduces the number of data planes (reservation cycles) needed to achieve some level of CLP. From Figure 6, a 256-input unbuffered PFDB requires 5 data planes to achieve a CLP below 10^{-6}, while the same level of performance can be achieved by using only 4 data planes if 2 buffers were used. When input buffering is used, the number of data planes needed to achieve a CLP of nearly 10^{-8} gradually decreases from 4 to 2 with increasing the buffer size from 2 to 10 as shown in Figure 7 for the case of a switch size of 256. This Figure shows the profitability of increasing input buffer size on a 256 × 256 PFDB.

4.2 Performance of pipelined partially-dilated banyans

Pipelined banyan uses simple banyan in control plane. Its throughput ranges from 0.65 to 0.25 for 32-input to 1024-input switches respectively. To reduce the number of data planes required to achieve some loss rate in pipelined architectures, one may use higher throughput banyan switchs such as partially dilated banyan $DB(n, d, 1)$.

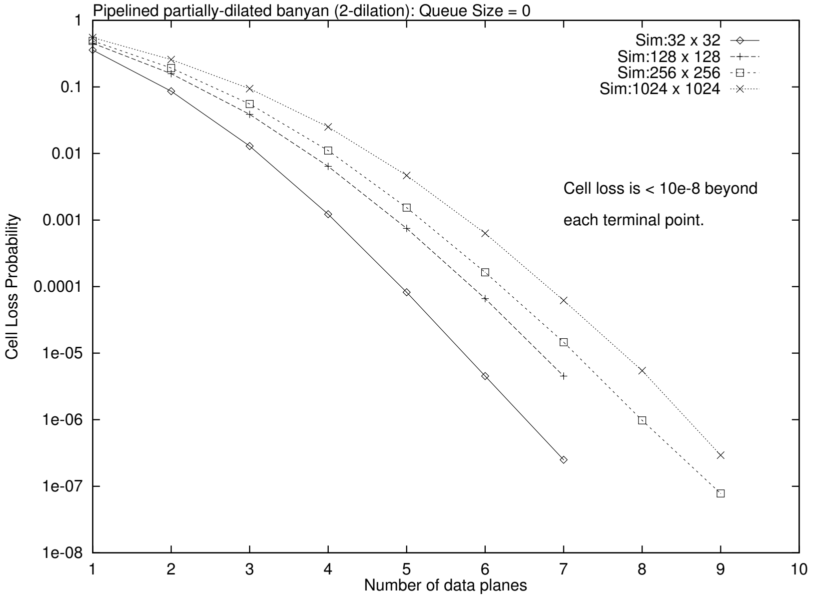


Figure 8: CLP of unbuffered pipelined partially-dilated (2) banyans

In this section we present the simulation results of *Pipelined Partially-Dilated Banyan* (PPDB) which uses one $DB(n,d,1)$ in each data plane. We have simulated the $PPDB$ for $d=1$ (2-dilation) and $d=2$ (4-dilation). There are 2 output ports (4) for each output channel in the case of 2-dilation (4-dilation).

Figures 8 and 10 show the loss rate of $PPDB$ with 2-dilation. Figures 9 and 11 show the loss rate of $PPDB$ in the case of 4-dilation. With no input buffering the 2-dilated $PPDB$ requires about 9 data planes against 12 for the pipelined banyan (1-dilated) as as shown on Figure 12. The profitability of dilation is even more significant in the case of 4-dilation which reduces the number of data planes to about 6 as shown in Figure 11.

The CLP reduces further when input buffers are used. We observe a similar trend as in unbuffered pipelined switches. A 2-dilated $PPDB$ with 10 input buffers requires only 2 data planes to achieve a loss rate below 10^{-6}. Notice that switching delay of partially-dilated banyan ($d>1$) is shorter than that of simple banyan ($d=1$) because of the expansion phase. Therefore, the use of partially-dilated banyans allows reducing the number of data planes in the pipelined switch while dropping also the reservation time compared to those of simple banyan. This reduction in the number of data planes and delay in pipelined partially-dilated banyan is gained at the cost of increased hardware and number of interconnection links in both control and data planes. Analysis of gained performance and cost will be presented at the end of this section.

4.3 Performance of pipelined banyans

In this section we present simulation results of *pipelined banyan* (PB) switches which refers to the scheme presented in [1]. Notice that the simple banyan switch is a particular case of ($DB(n,0,1)$) for which the dilation degree $d=0$ and the number of sorter stages in each *D-SW* is 1.

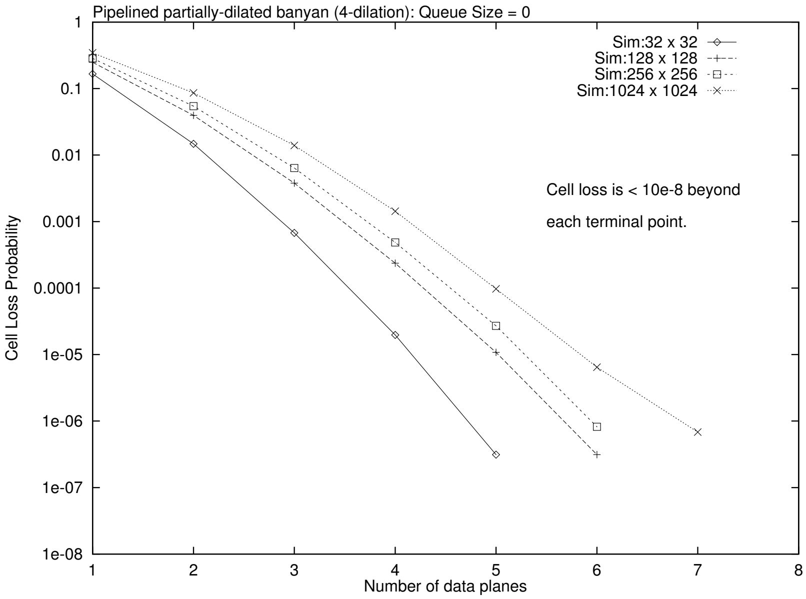


Figure 9: CLP of unbuffered pipelined partially-dilated (4) banyans

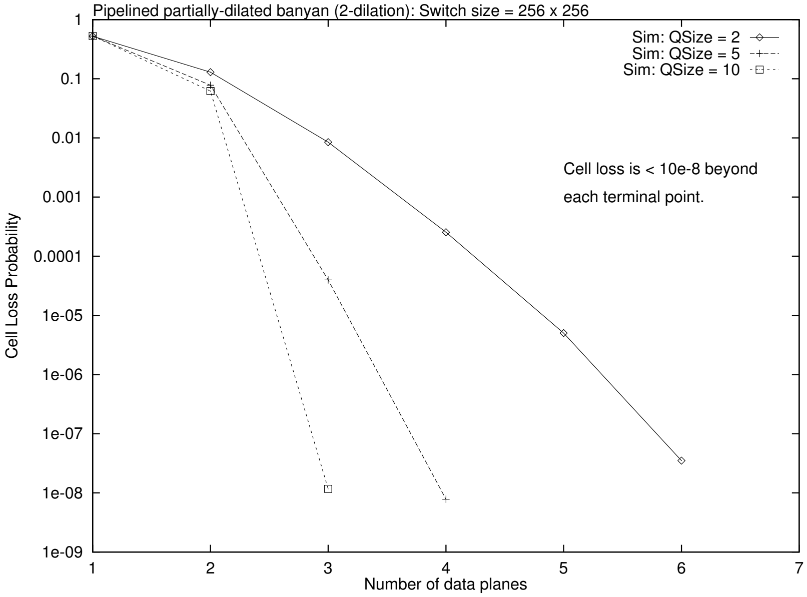


Figure 10: CLP of buffered pipelined partially-dilated (2) banyans

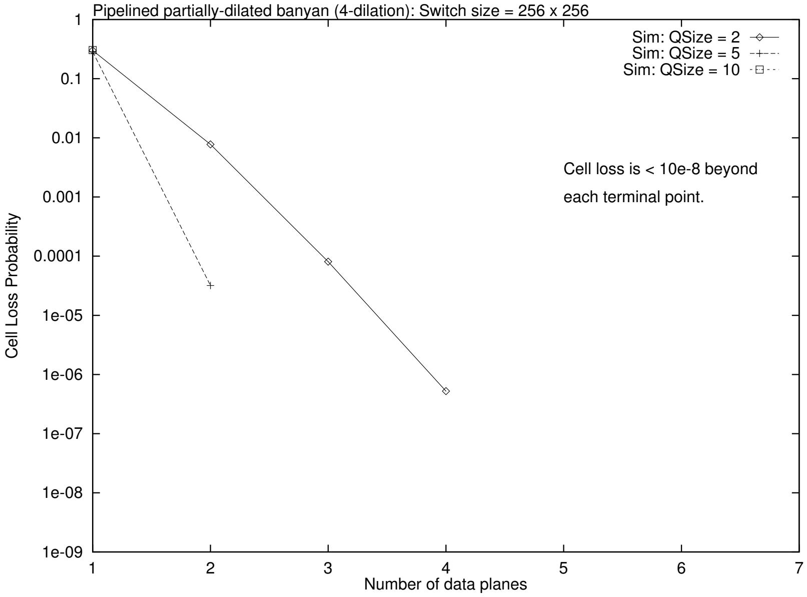


Figure 11: CLP of buffered pipelined partially-dilated (4) banyans

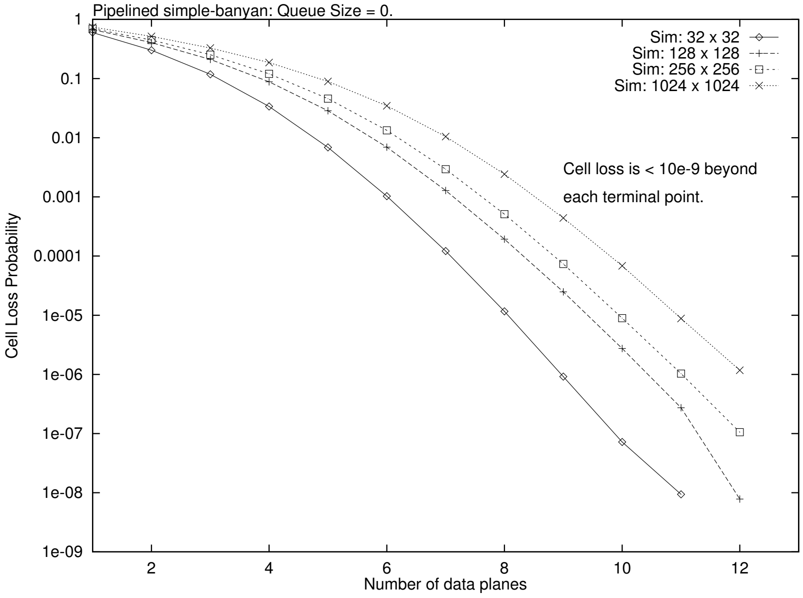


Figure 12: CLP of unbuffered pipelined banyans

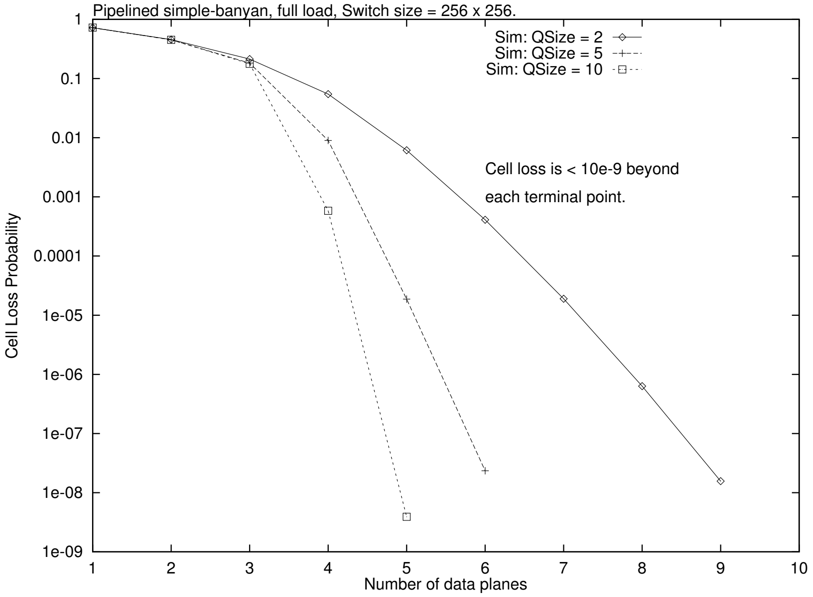


Figure 13: CLP of buffered pipelined banyans

In unbuffered PB the loss can occur in the last reservation slot, while in buffered PB the loss occurs at input of full buffers. Figure 12 shows the CLP as function of the number of data plane for unbuffered PB. To achieve a CLP below 10^{-6} the number of data planes required is 12 for large switches such as 256×256 or 1024×1024. The number of data planes required can be reduced when input buffers are available. In buffered PB, a cell that fails in a reservation slot remains in the input buffer until it succeeds in some subsequent reservation slot. Figure 13 shows the profitability of input buffering for pipelined simple banyan. This Figure shows the effect of varying input buffer size for a 256×256 switch. When the input buffer size is increased from 2 to 10 we observe a decrease in the number of data planes from 9 to 5 while achieving a CLP around 10^{-8}.

Though the simple banyan has relatively poor passthrough input buffering contributes significantly in boosting the performance of the pipelined switch. Our simulation indicates that six or seven data planes are required to achieve a CLP below 10^{-6} for large switches with 10-input buffering.

4.4 Queuing delays

In this section we study HOL delays as well as total switch delays. The HOL delay is counted from the time a cell becomes HOL to the time the cell succeeds in making a reservation. Therefore, the HOL delay is nil when a cell makes a reservation from the first reservation attempt. The delay is expressed in terms of reservation cycles. The total switch delay is counted from the time a cell enters some input buffers until the time at which the cell succeeds in making a reservation.

Figures 14 and 15 show the average HOL delay and average total delay for the PB, $PFDB(2)$ with 2-dilation, $PPDB(2)$ with 2-dilation, and $PPDB(4)$ with 4-dilation when 2 input buffers are available for each input port. Lighter input loads certainly contribute

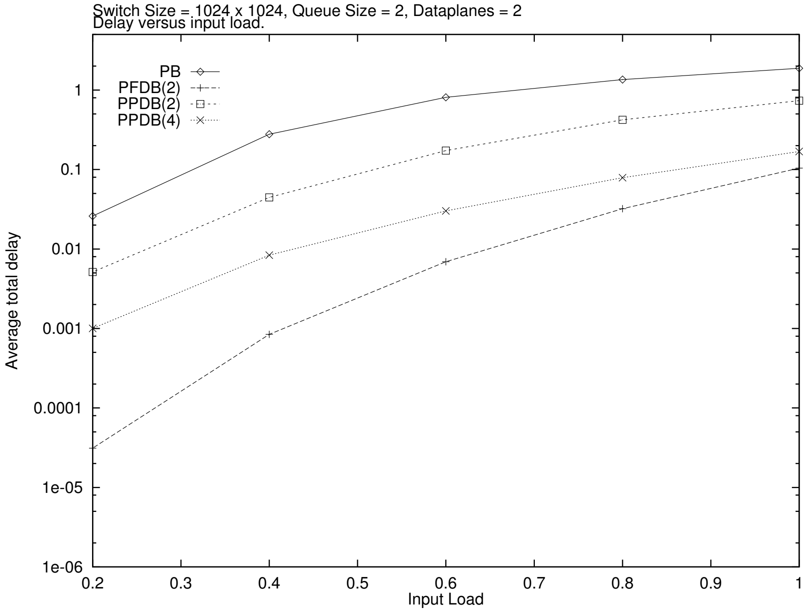


Figure 14: Comparison of Head-of-Line delay

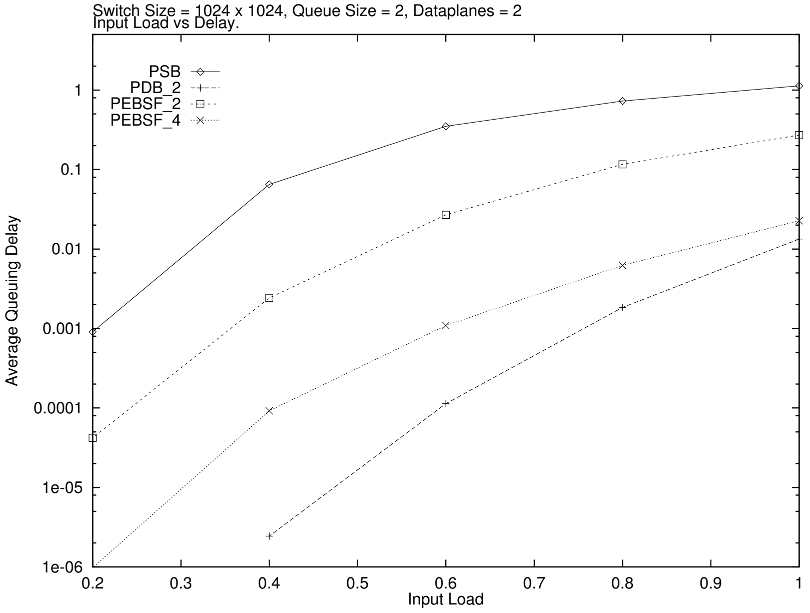


Figure 15: Comparison of total queuing delay

	PB	PDB_2	PDB_4	PDB_8	FDB_2	FDB_4	FDB_8
$DB_{Sorters}$	2048	3584	6144	10240	7168	24576	81920
DB_{Dmux}	2048	3840	6912	12032	3840	6912	12032
DB_{Link}	6400	11520	20224	34560	18688	57088	177920
$\tau_{control}$ (in Gate delay)	184	172	160	148	201	321	513
τ_{data} (in Gate delay)	24	22	20	18	36	56	88

Table 2: Hardware requirements and delays for 256 × 256 dilated banyans

in shortening the HOL delay and total delay for all switches. Mainly, delays are important when input traffic is close to full load. Pipelined switches employing $PFDB(2)$ or $PPDB(4)$ have high throughput which indicates that a cell remains, on the average, HOL for small fraction (about 10^{-2}) of a reservation cycle at full load. At full load, the throughput of PB and $PPDB$ (2-dilation) is much less compared to the other switches which causes an HOL delay of 10^{-1}. The total delay (Figures 15) for PB and $PPDB(2)$ with the same conditions becomes closer to 1 reservation cycle. In the case of PB and $PPDB(2)$ at full load, a cell may: 1) waits in input buffer for 1 clock before becoming HOL, or 2) finds the buffer empty and succeeds in the reservation attempt at second cycle. In this case, input buffering is profitable and buffer size significantly affects CLP.

In the next section we compare the CLP for the above pipelined switches together with their hardware requirements, costs, propagation delays, and service rates.

4.5 Comparisons

The performance of pipelined switches cannot be characterized only by the achieved throughput or the corresponding CLP. The time it takes to achieve a level of performance is one important factor in the evaluation. For example, the pipelined banyan can achieve a CLP below 10^{-6} when there are six data planes. Whether this performance is adequate for ATM rates or not depends on: (1) the number of needed reservation slots (6), (2) switching delay in control plane, and (3) payload transmission time in data plane. Using the results from Section 3.2 for a 256-input dilated switch we list the hardware requirements in Table 2. To compare performance of the pipelined switches that employ DBs we need to evaluate the service rate as function of gate delays. The service rate of the switch is the number of cells that can be switched per unit of time. Though wires occupy significant space and cause complexity in VLSI and PCB we do not account for wiring and wire delays for simplicity. This assumes that delays through a logical gate in VLSI dominate delays on wire.

A payload of N_{cell} bits can be transmitted over a free data plane in $T_{data} = N_{cell}\tau_{bit} + \tau_{data}$ gate delays, where τ_{data} is the delay in one data plane and τ_{bit} is the time in gate delays to remove one bit from input buffer, i.e. $1/\tau_{bit}$ is rate of payload transmission. In the control plane, the switching time (also reservation slot) of the header is $T_{control} = \tau_{control}$ gate delays which accounts for self-routing of the header along a path from input to output. Note that $N_{dp} = \lceil T_{data}/T_{control} \rceil$ is the least number of needed data planes in the pipelined switch to guarantee there is a free data plane for payload transmission at the

Buffered pipelined multi-plane switches (input buffer size is 10)

	PB	$PPDB_2$	$PPDB_4$	$PFDB_2$
Number of reservation slots L	6	3	2	2
Achieved CLP	4×10^{-9}	1.2×10^{-8}	2×10^{-8}	1.2×10^{-8}
T_{data} (in Gate delay)	448	446	444	460
$T_{control}$ (in Gate delay)	184	172	160	201
$N_{dp} = \lceil T_{data}/T_{control} \rceil$	3	3	3	3
$N_{dp}^{Phys} = Min\{N_{dp}, L\}$	3	3	2	2
Gates in control plane	65536	117504	205056	192768
Gates in N_{dp}^{Phys} data planes	49152	87552	152064	101064
Links in control and N_{dp}^{Phys} data planes	25600	46080	60672	56064
Gates in pipelined switch	114688	205056	306432	294144
Service Rate (Gega Cells/s)	0.23	0.5	0.8	0.64

Table 3: Service rates of 256 × 256 pipelined dilated banyans

end of each reservation slot.

Denote by p the cell input load and let clp be the overall CLP when each time slot is formed by L reservation slots. In steady state, the average number of cells that can be switched in one time slot is $p(1 - clp)2^n$ for 2^n-input switch. For pipelined switches each time slot consists of L reservation slots but this requires $N_{dp}^{phys} = Min\{N_{dp}, L\}$ physical data planes. The duration of one time slot is then $T = L\tau_{control}$ gate delays. The time to switch $p(1 - clp)2^n$ cells is T gate delays. If one assumes N_{dp}^{Phys} data planes, then the service rate (S) of the pipelined switch will be:

$$S = \frac{p(1 - clp)2^n}{L\tau_{control}}$$

We assume header and payload are stored into the input buffers and retrieved at a rate of *one bit per gate delay* ($\tau_{bit} = 1$). One gate delay is assumed to be 1 ns. Now we consider a number of buffered pipelined DBs which are PB, $PPDB_2$, $PPDB_4$, and $PFDB_2$ and list some of their parameters in Table 3. L is being the number of reservation slots required to achieve a CLP below 10^{-8}.

For PB the number of needed reservation slots is $L = 6$. Since $N_{dp} = 3$ we can make 6 reservation cycles by using 3 data planes. The same considerations apply to the other switches. Due to low throughput of simple banyan, PB requires higher number of data planes than a pipelined DBs which requires between 2 and 3 more complex data planes.

The service rate is evaluated for PB, $PPDB_2$, $PPDB_4$, and $PFDB_2$ with 256 inputs and outputs. Since the number of reservation slots L is dictated by the need for an acceptable CLP. Pipelining allows minimizing the cost of L reservation slots through the use of only N_{dp}^{Phys} data planes. Therefore, the service rate S is one *structural feature* of the used banyan which cannot be increased beyond some limit for a given design technology. The service rate of PB is the least for the family of pipelined DBs. According to our design model and assumptions, a 256-input PB can switch cells with a CLP below 10^{-8}

Type of Source	Mean burst size in cells	Average bit rate $SCR \times 384$ *bps*	Burstiness β	Cell Loss Tolerance
CBR (Voice)	N/A	64 Kbps	1	10^{-4} to 10^{-6}
Connectionless data	200	700 Kbps	as high as 1000	10^{-12}
Connection oriented data	200	25 Mbps	as high as 1000	10^{-12}
VBR video	2	25 Mbps	2 to 5	10^{-10}
Background data/video	3	1 Mbps	2 to 5	10^{-9} to 10^{-10}
VBR video/data	30	21 Mbps	2 to 5	10^{-9}

Table 4: Types and features of some ATM traffics

only when overall cell arrival rate is below 230×10^6 cells/s. This represents a structural limit of pipelined banyans.

One way to achieve higher service rates than that of PB is to use DBs. The highest service rate is achieved for $PPDB_4$ which indicates that a 4-dilation with one-stage of sorters in the D-SW switch was critical in providing high throughput without dramatically increasing the reservation time. It is shown in Table 3 that a $PPDB_4$ has nearly four times the service rate of PB at hardware cost of nearly three times that of PB. There are two reasons for the relatively high service rates of of partially-dilated banyans. First, the non-blocking binary expansion phase is responsible for significantly increasing the rate of successful reservations compared to simple banyans. Second, the use of one sorter stage in each D-SW switch of partially dilated banyans was the key factor to maintain a low propagation delay compared to fully-dilated banyans. One can think of partially-dilated banyans as a banyan whose service rate can be scaled up by horizontally expanding the banyan architecture (increasing dilation) without increasing the path delay (stages of D-SW).

In the next section we study the performance of the above pipelined switches under ATM traffic which will allow us to comments on the robustness of each switch.

5 Performance under ATM traffic conditions

ATM networks are being engineered to support bursty and non-bursty sources with a wide range of bandwidth requirement. The communication services provided at the ATM layer consist of the following five service categories [16]: (1) Constant Bit Rate (CBR), (2) Real-Time Variable Bit Rate (rt-VBR), (3) Non-Real-Time Variable Bit Rate (nrt-VBR), (4) Unspecified Bit Rate (UBR), and (5) Available Bit Rate (ABR). See Table 4 for other features. The CBR and rt-VBR services are for real-time sources with hard cell delay and delay jitter requirements and limited tolerance to cell loss (ex: audio and video sources). At connection establishment, a source must declare its Quality of Service requirements (QoS). For CBR sources, the only QoS parameter required is the Peak Cell Rate (PCR). For rt-VBR, both the PCR and the Sustainable (average) Cell Rate (SCR) are required.

The remaining service categories (nrt-VBR, UBR, and ABR) are for bursty non-real-time traffic sources. The nrt-VBR service is for applications with loose cell delay and delay jitter requirements and low cell loss (such as voice mail and some video applications). In

Traffic	Source type	Peak arrival rate (Mbps)	Average cell arrival (Mbps)	Mean burst length (cells)	Burstiness	Percentage channels
Traffic 1	CBR	0.064				10%
	CBR	1.4				10%
	VBR		0.7	200	5	20%
	VBR		25	20	5	20%
	VBR		21	30	4	40%
Traffic 2	CBR	0.064				25%
	CBR	1.4				25%
	VBR		0.7	200	5	12%
	VBR		20	25	5	13%
	VBR		2	25	10	6%
	VBR		3	1	5	6%
	VBR		30	21	4	6%
	VBR		3	6	5	7%

Table 5: Features of *traffic 1* and *Traffic 2*

addition to the PCR and SCR, nrt-VBR sources must specify their Maximum Burst Size (MBS). The UBR service category is a best effort service. Sources using this service are not required to specify any QoS parameters (ex: connection-less data). Finally, for ABR traffic sources, the application is required to specify both its PCR and its Minimum Cell Rate (MCR). Examples of applications that may use this service are file transfer, email, LAN Emulation, etc.

A realistic ATM workload is a mixture of bursty and non-bursty sources with the load originated from a variety of traffic sources which exhibit correlation in space as well as in time. Traffic source characterization has been an extensive area of research[17]. A simple and widely adopted traffic source model is the *ON-OFF model*. According to this model, during the lifetime of a virtual connection, the traffic source will be in one of two states, *active* or *idle*. During the active state the source is transmitting cells at some given rate. Each active state may be followed by an idle period during which the source is silent. The cells generated during the same ON-period form a *burst*. Furthermore, it is always assumed that successive active and idle periods are statistically independent and exponentially distributed. As suggested by ITU-T, the length of the active period as well as that of the idle period are exponentially distributed.

For simulation purposes, several parameters have been identified, which together, completely characterize an ON-OFF traffic source. These are, the PCR, the SCR, and the average duration of the ON-state (t_{on}). Other parameters of interest such as the source burstiness (β) or the average duration of the OFF-state (t_{off}) are easily derived from these three parameters. For example, the $\beta = \frac{PCR}{SCR}$ and $t_{off} = (\beta - 1)t_{on}$. Typical values for the traffic parameters for examples of traffic sources are summarized [17].

In our simulation study, we assumed that the PCR, t_{ON}, and β are known for each source. Furthermore, as recommended by ITU-T, we assumed that the active and idle periods are exponentially distributed with parameters $a = \frac{1}{t_{ON}}$ and $b = \frac{1}{t_{OFF}}$ respectively. We experimented with the following traffic mixes. *Traffic 1* and *Traffic 2* for which the source types are given in Table 5.

To generate traffic sources according to the ON-OFF model VBR traffic sources require specification of four input parameters m, B, and β in addition to percentage of channels. While for CBR sources the same generation can be done when only the peak arrival rate p and the percentage of channels are given.

We subjected the pipelined banyan PB, pipelined partially-dilated banyan $PPDB$, and pipelined fully-dilated banyan $PFDB$ switches to *Traffic 1* and *Traffic 2*. For *Traffic-1*, it was observed that less number of data planes were required to achieve some CLP than that needed in the case of uniform traffic for equal number of input buffers. The above observation was true for all three switches and for all switch sizes.

The reason is that traffic load under *Traffic-1* and *Traffic-2* causes less conflicts compared to full load uniform traffic which explains why less number of data planes are needed for achieving the same CLP. Figures 16 shows the CLP for $PFDB$ switch with 2-dilation and 2 input buffers under *Traffic-1.* Two data planes were required to acheive a CLP level close to 10^{-6} for $PFDB$ and between 3 and 4 for $PPDB$ as shown on Figure 17. Using the same number of data planes, the CLP of $PFDB$ dropped below 10^{-8} for all switch sizes when the buffer size was increased to 5.

We repeated the above experiments by using *Traffic-2* with 2 input buffers and noticed that the CLP is much less than that obtained under *Traffic-1* for the same number of data planes. Specifically, the CLP of PB, $PPDB$, and $PFDB$ was below 10^{-8} when 2 data planes and two or more input buffers were used with all switch sizes. The reason is that in *Traffic-2* only 50% of the sources are VBR sources compared to 80% in the case of *Traffic-1.*

Figures 18 show the CLP for $PPDB$ with 4-dilation when using 2 input buffers are used under *Traffic-1* which can be compared to the case of 2-dilation shown on Figures 17. Increasing the dilation degree of $PPDB$ under *Traffic-1* leads to lower CLP or lesser number of needed data planes which indicates that CLP is significantly affected by the degree of dilation under both uniform and ATM traffic. Figures 19 show the effect of varying the input buffer size for $PPDB$ with 2-dilation under *Traffic-1.*

Figures 20 and 21 show the CLP for PB when using 2 input buffers under *Traffic-1* and *Traffic-2*, respectively. In the case of *Traffic-2*, much less number of data planes are needed to achieve equal level of CLP which indicates that PB is also very sensitive to percentage of VBR sources. Figures 22 and 23 show that increasing the buffer size can be rewarded by a significant drop in the number of data planes needed to achieve a CLP of 10^{-6} or less. For example, the number of data planes dropped from 6 to 4 when increasing input buffering from 2 to 10 in the case of a 1024-input PB. We bellieve that increasing buffer size within some reasonable limits is always easier to manage than increasing the number of data planes required to achieve some level of performance.

6 Conclusion

In this paper we evaluated a pipelined switch architecture which uses dilated banyans to achieve high service rate. Since ATM rates are on the order of Gbps, the switch service rate should be at the same level as cell arrival rates to avoid buffer overflow.

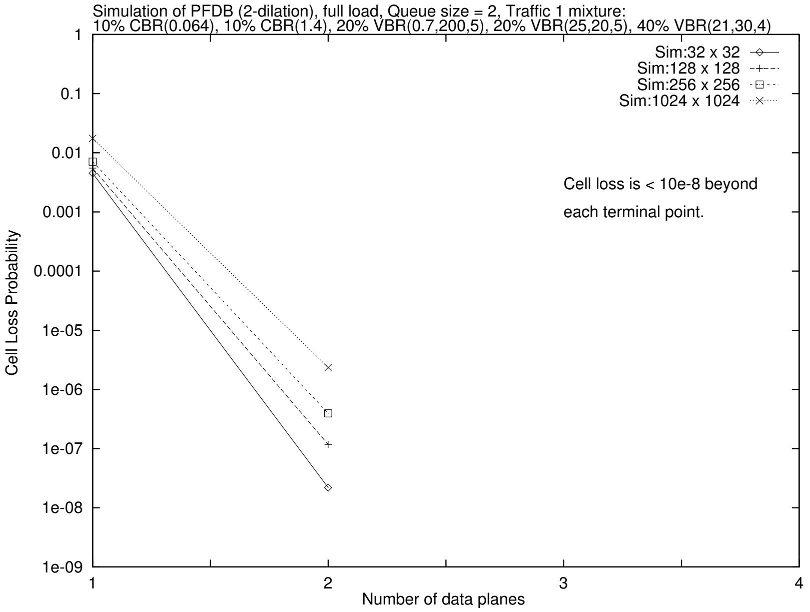


Figure 16: PFDB with 2-dilation under ATM *Traffic-1*

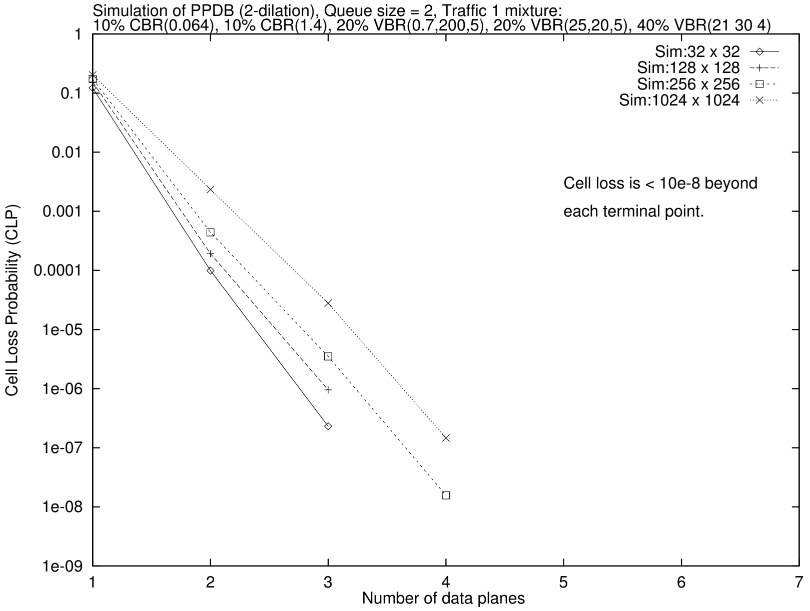


Figure 17: PPDB with 2-dilation under ATM *Traffic-1*

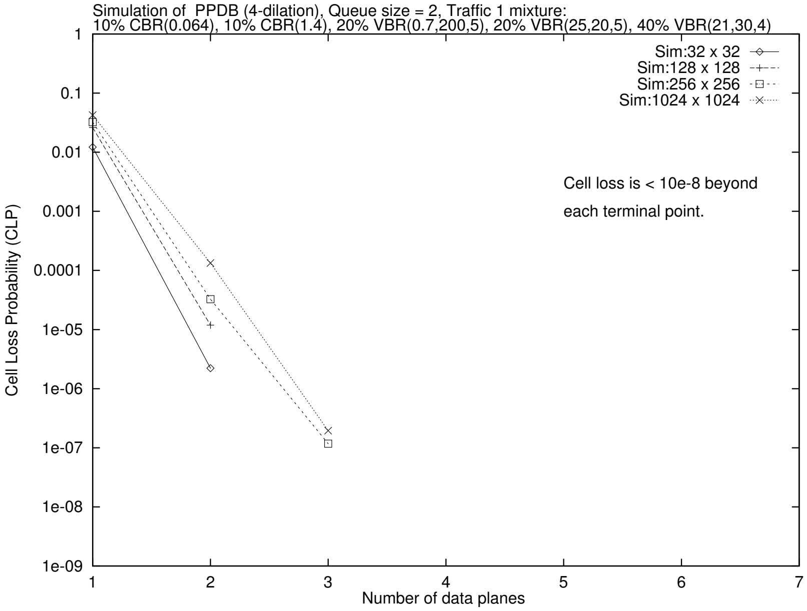


Figure 18: PPDB with 4-dilation under ATM *Traffic-1*

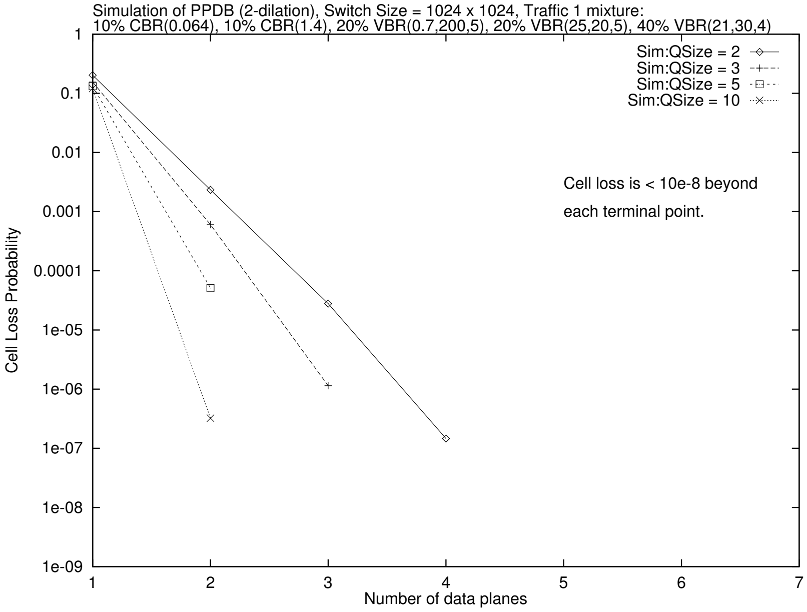


Figure 19: Effect of increasing buffer size on 1024-input PPDB(2) under *Traffic-1*

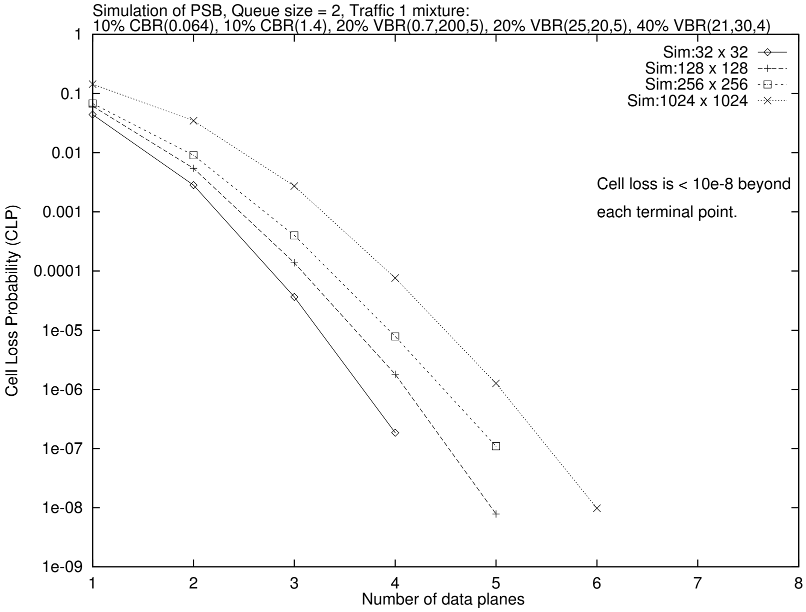


Figure 20: CLP of PB with 2 input buffers under *Traffic-1*

The pipelined scheme has been previously proposed for simple banyans. We proposed a class of pipelined ATM switches based on dilated banyans for which the throughput can be scaled up without dramatically incrasing of the switching time. We simultated the proposed dilated banysns by using the pipeliened scheme and showed that it can deliver a high service rate. To compare switching delay of different banyans we proposed a modular self-routing hardware for the design of specific dilated banyans from where to eatimate the switching delay. processing overhead. This allows engineering the design of high throughput pipelined switches with respect to hardware complexity, cell switching delay, interconnection links, or a combination of the above. Evaluation of pipelined dilated banyans was carried out under uniform traffic as well as under simulated ATM traffic mixes. We compared the obtained service rates and CLPs of the pipelined scheme for the case of simple banyan and dilated banyans. The above switches were also compared with respect to hardware requirements, buffering and switching delay, number of needed interconnections, CLP. We also studied the robustness of the proposed pipelined switches under a variety of ATM traffic conditions.

7 Appendix

Harware complexity

We evaluate the hardware requirements of the $DB(n, d, k)$, number of interconnection links, and time delay through a path from input to output.

We show that bandwidth of $D\text{-}SW(d, k)$ (2 input and output channels) scales up with increasing the number of sorter stages k up to $k = 2^d$.

Lemma 1 *In 1:2^d dilated banyan, there must be 2^d sorter stages in the concentrator of each $D\text{-}SW(d, k)$ if no cell loss should occur within the concentrator when at most 2^d cells*

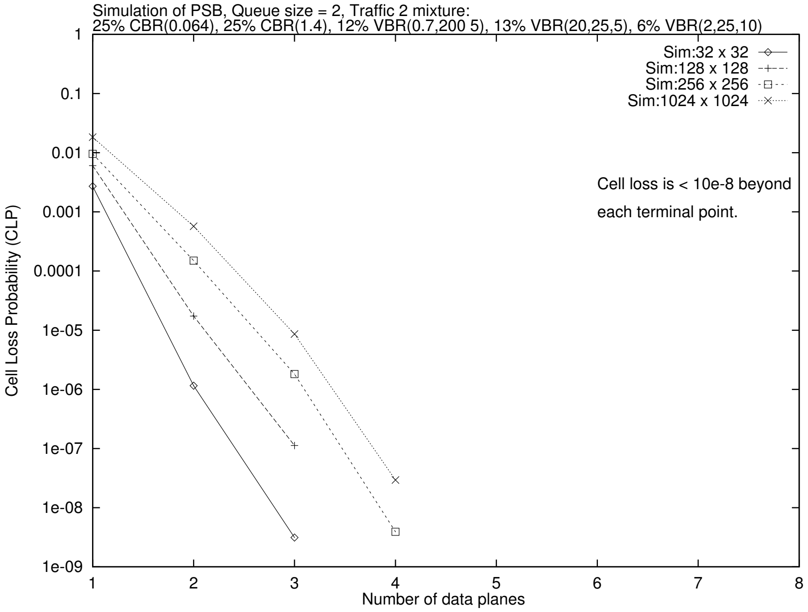


Figure 21: CLP of PB with 2 input buffers under *Traffic-2*

are present on the 2^{d+1} *input ports of concentrator.*

Proof An upper (lower) output of up-sorter (down-sorter) in first stage transmits a cell if there is at least one cell at input of concentrator. However, the lower (upper) output of the same up-sorter (down-sorter) transmits a cell if at least 2 cells are present at input of concentrator. An up-sorter (down-sorter) in the ith stage receives: 1) a cell on one of its inputs if there is one cell or more at input of concentrator, and 2) a cell on the other input if there is at least i cells at input of concentrator. In the example shown on Figure 3-(c), we marked on input links the minimum number of cells arriving on concentrator input so that at least one cell is received at the corresponding input link. Since the lower output has least priority, the lower (upper) outputs of ith stage up-sorter (down-sorter) transmits a cell if and only if there is at least $i+1$ cells at input of concentrator. There are 2^{d+1} inputs ports and 2^d output ports for each of UC and LC. Therefore, there must be 2^d stages in each of the UC and LC to guarantee that no cell loss can occur in last stage of concentrator as long as the number of input cells is no more than 2^d at concentrator input. ■

We now evaluate the number of needed demultiplexers and sorters in an n-stage $DB(n,d,k)$.

Lemma 2 *A 1:*2^d *n-stage* $DB(n,d,k)$ *has* $DB_{sorter}(n,d,k) = k(n-d)2^{n+d}$ *Sorters and* $DB_{Dmux}(n,d,k) = 2^{n+d}(n-d+1) - 2^n$ *Demultiplexers.*

Proof To expand one input into 2^d we need $2^d - 1$ demultiplexers DMs distributed as a binary tree over the d stages. For all 2^n inputs of the expansion, the number of DMs must be $2^n(2^d-1)$. In the routing phase, each *D-SW* switch has 2^{d+1} inputs and there is one DM per input. The total number of DMs in one stage of the routing phase

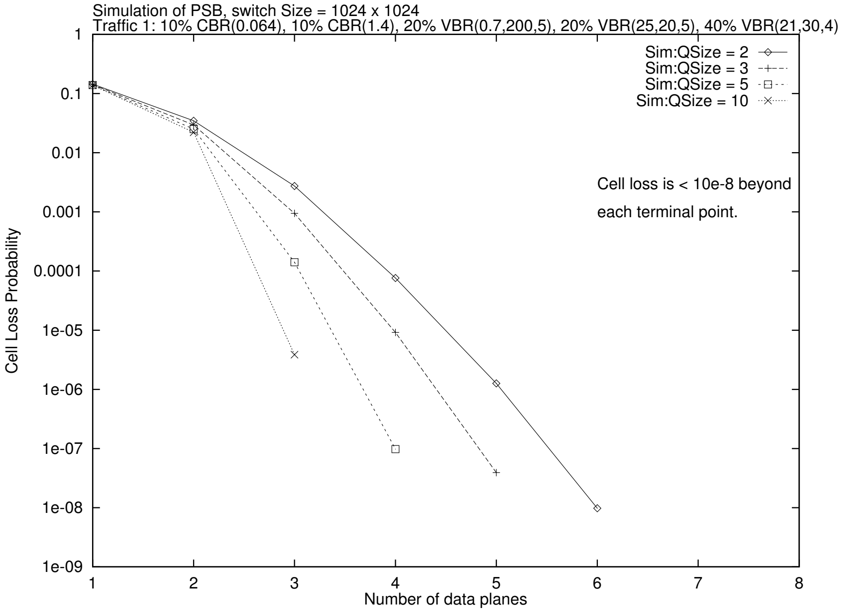


Figure 22: Effect of increasing buffer size on 1024-input PB under *Traffic-1*

is $2^{d+1} \times 2^{n-1}$ and $2^{d+1} \times 2^{n-1}(n-d)$ for all the $n-d$ stages. The total number of DMs is then $2^{n+d}(n-d+1) - 2^n$.

Each $D\text{-}SW(d,k)$ switch in the routing phase is k-stage and there are 2^{d+1} sorters in each stage. Therefore, we have $k \times 2^{d+1}$ sorters in each $D\text{-}SW(d,k)$ switch. There are 2^{n-1} $D\text{-}SW(d,k)$ in each stage of the routing phase. For all $n-d$ stages we have $k2^{d+1}2^{n-1}(n-d)$. In all the $n-d$ stages of the routing phase we then have $k2^{d+n}(n-d)$ DMs. The expansion has no sorters, then the total number of sorters in an $DB(n,d,k)$ is $DB_{sorter}(n,d,k) = k(n-d)2^{n+d}$ DMs. ■

We now evaluate the number of needed interconnections in an n-stage $DB(n,d,k)$.

Lemma 3 *A $DB(n,d,k)$ made of demultiplexers and sorters has $DB_{Link}(n,d,k) = 2^{n+d}$ $[(n-d)(2k+1)+2] - 2^n$ interconnection links.*

Proof $DB_{Link}(n,d,k)$ is the sum of number of links in expansion and routing phase. Each link at input of the expansion is doubled at each stage. After d stages, each link get expanded into 2^d and the total number of links is $2^{d+1}-1$. For all 2^n input links, the total number of links in the expansion is then $2^n(2^{d+1}-1)$.

The routing phase consists od $n-d$ stages each has 2^{n-1} $D\text{-}SW$ switches. Each $D\text{-}SW$ has 2^{d+1} inputs and outputs. The first stage demultiplexer of $D\text{-}SW$ double the links to 2^{d+2}, of which 2^{d+1} go the UC and the other 2^{d+1} go to LC. There are k stages (sorter) in each of UC and LC with a total of $k \times 2^{d+2}$ interstage links and 2^{d+1} output links. The total number of links in the $D\text{-}SW(d,k)$ is $2^{d+1}(2k+1)$. For all the stages of the expansion we have $(n-d)(2k+1)2^{n+d}$ links. Therefore, the total number of links in an n-stage DB is $2^n(2^{d+1}-1) + (n-d)(2k+1)2^{n+d}$ which simplifies to $DB_{link}(n,d,k) = 2^{n+d}[(n-d)(2k+1)+2] - 2^n$. ■

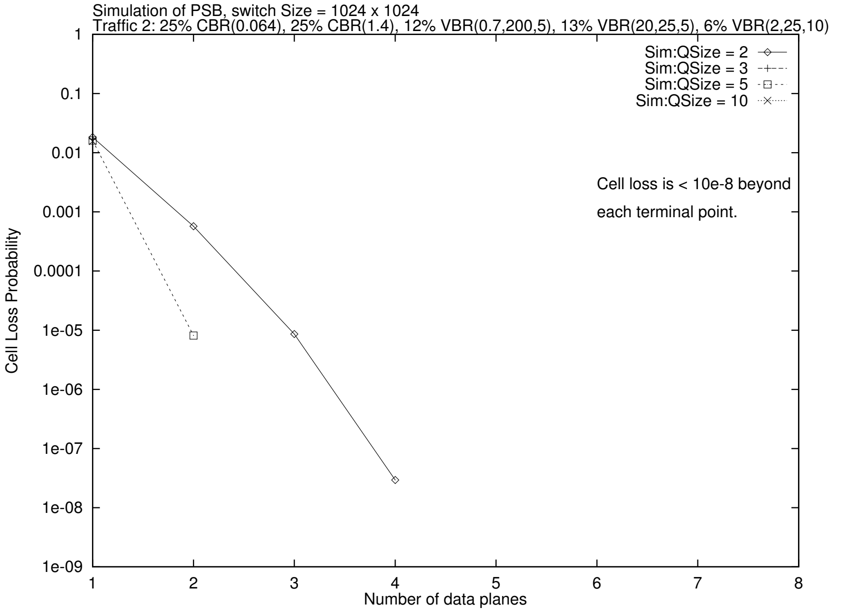


Figure 23: Effect of increasing buffer size on 1024-input PB under *Traffic-2*

We now evaluate the *switching delay* in control plane that is the time needed for self-routing of the cell header along a path from input to output of $DB(n, d, k)$. We also need to evaluate the propagation delay in data plane.

Lemma 4 *A $DB(n, d, k)$ is characterized by (1) a switching delay $\tau_{control}(n, d, k) = [11n + 12k(n - d)]\tau_{gate}$ in control plane, and (2) a propagation delay $\tau_{data}(n, d, k) = [n + 2k(n - d)]\tau_{gate}$ in the data plane, where τ_{gate} is one gate delay time.*

Proof The total number of DMs along one path from input to output is n because a path traverses only one DM in each stage of the expansion and routing phases. The same path traverses one *D-$SW(d, n)$* switch in each of the $n - d$ stages of routing phase. Denote by τ_{Dmux} and τ_{Sorter} the delays through a DM and a sorter, respectively. A path through a *D-$SW(d, k)$* encounters k sorters, then the total delay due to sorters in the routing phase is $k(n - d)\tau_{Sorter}$. The overall delay in DB due to expansion and routing phases is then $DB_\tau(n, d, k) = n\tau_{DM} + k(n - d)\tau_{Sorter}$.

A sorter needs a D-latch to store the state and two parallel 2×1 multiplexers. Latching of the presence bit requires two gate levels. The state is function of the two priority bits and two presence bits of the two input cells. Using dual input gates, the number of gate levels required to implement the sorter state as sum of products is $log_2(n_v 2^{n_v})$, where n_v is number of Boolean variables per minterm. Therefore, the number of gate levels in a state function with 4 variables is $log_2(4 \times 2^4) = 6$ levels. Two gate levels are needed in each of D-latch and the 2×1 multiplexer. Hence $\tau_{Sorter} = 12\tau_{gate}$, where τ_{gate} is one gate delay.

A DM requires 11 gate levels ($\tau_{DM} = 11\tau_{gate}$) because it needs 2 gate levels to latch the presence bit, 6 gate levels for the state (up or down), 2 gate levels to latch the state, and

1 gate level for the 1 × 2 demultiplexer. Therefore, the delay in control plane expressed as function of number of gate delay is $\tau_{control}(n, d, k) = [11n + 12k(n-d)]\tau_{gate}$.

In the data plane paths from input to output are pure combinational logic paths for which routing is established based on previously achieved switching in the control plan. The switch states from control plane are used for setting the route in the data plan. The delay in control plane was found to be $DB_{\tau}(n, d, k) = n\tau_{DM} + k(n-d)\tau_{Sorter}$. Each sorter in control plane is associated in the data plane two parallel 2 × 1 MUXs each requires 2 gate level delays. Each DM in control plane is associated in data plane one 1 × 2 DMUX with 1 gate delay. Therefore, the total delay of the combinational logic path from input to output in the data plane is $\tau_{data}(n, d, k) = [n + 2k(n-d)]\tau_{gate}$. ■

Analytical model of pipelined DB

The analytical model of pipelined DB uses the queing model of [1] which requires the knowledge of passthrough probability of $DB(n, d, k)$ as function of input load. We derive analytical expression for the passthrough of $DB(n, d, k)$. The workload assumptions are as follows. All switch inputs are identical and independent. In every time slot, each input port of $DB(n, d, k)$ has a probability p of having a cell and $1-p$ of having no cell. We also assume that the activity level at the various input ports are independent in time as well as in space. The cell destinations are uniformly distributed over all the switch outputs. The results are obtained while adopting an Omega topology for the banyan networks.

A $DB(n, d, 2^d)$ has d expansion stages and $n-d$ routing stages. We assume uniform cell rate issue on DB with probability q. Each output port out of 2^{n+d} ports of expanssion phase has a probability of $q2^{-d}$ to transmit a cell. The passthrough probability of the expanssion is $B_{expansion} = q2^{-d}$. To evaluate the passthrough of the routing phase we find the passthrough (B_{DSW}) for $D\text{-}SW(d, 2^d)$ which can be recursively used along the $n-d$ stages to obtain overall passthrough of DB.

The D-SW switch uses cell destinations to partition incomming cells into: (1) set G_0 formed by input of UC, and (2) set G_1 formed by input of UC. Each set at most has 2^{d+1} cells. Next, each group is sorted in the decreasing order of priority. At most, the top 2^d cells of each group are switched to the 2^d output ports of UC or LC. In last stage of concentrator, a cell loss occurs when either G_0 or G_1 contains more than 2^d cells. In last stage a sorter has two inputs but only one output. The $D\text{-}SW(d, 2^d)$ switch can pass at most 2^d cells on its upper or lower output channel. Cells in excess of the permissible 2^d are lost.

Assuming uniform cells rate issue on the input ports of D-SW with probability p and let $B(p)$ be the probability a cell to traverse the switch. The average number of cells that traverse the switch in one time slot is $B(p) \times 2^{k+1}$. The probalility there is a cell on a given input port on either UC or LC is $p/2$. The probability there are i cells on i input ports of UC and no cells on the remaining $2^{d+1} - i$ ports of UC is $(p/2)^i(1-p/2)^{2^{d+1}-i}$. Therefore, the probalility $P(i \,/\, 2^{d+1})$ that i cells are present on some input ports of either UC or LC among 2^{d+1} ports is:

$$P(i \,/\, 2^{d+1}) = \sum_{j=1}^{2^{d+1}} \frac{2^{d+1}!}{j!(2^{d+1}-j)!} \times (p/2)^j(1-p/2)^{2^{k+1}-j}$$

A $D\text{-}SW(d, 2^d)$ can pass at most 2^d cells through UC with 0 destination out of at most i present cells, where $0 \le i \le 2^{d+1}$. The cells in excess of 2^d that all have 0 destination are lost as there is no way to switch them out correctly. For this we need to consider the following three cases. First, we consider the case of j cells with 0 as destination out of i present cells and $1 \le j \le i \le 2^d$. All the j cells can be switched out. The term $B_{case-1}(p)$ is the passthrough for this case:

$$B_{case-1}(p) = \sum_{i=1}^{2^d} \frac{2^{d+1}!}{i!(2^{d+1}-i)!} \times (p/2)^i (1-p/2)^{2^{d+1}-i} \times \sum_{j=1}^{i} \frac{i!}{j!(i-j)!} \times \frac{j}{2^i}$$

Second, we consider the case j cells are present, $1 \le j \le 2^d$, with 0 destination out of i cells such that $2^d + 1 \le i \le 2^{d+1}$. Since the number of cells j requesting output UC satisfies $j \le 2^d$, then all j cells can also be switched. The term $B_{case-2}(p)$ is the passthrough for this case is:

$$B_{case-2}(p) = \sum_{i=2^d+1}^{2^{d+1}} \frac{2^{d+1}!}{i!(2^{d+1}-i)!} \times (p/2)^i (1-p/2)^{2^{d+1}-i} \times \sum_{j=1}^{2^d} \frac{i!}{j!(i-j)!} \times \frac{j}{2^i}$$

Third, we consider the case where there are j $(2^d + 1 \le j \le i)$ cells with 0 destination out of i cells such that $2^d + 1 \le i \le 2^{d+1}$. Since the number of cells j requesting output group UC exceeds 2^d, then only 2^d cells can be switched. The term $B_2(q)$ is the passthrough for this case is:

$$B_{case-3}(p) = \sum_{i=2^d+1}^{2^{d+1}} \frac{2^{d+1}!}{i!(2^{d+1}-i)!} \times (p/2)^i (1-p/2)^{2^{d+1}-i} \times \sum_{j=2^d+1}^{i} \frac{i!}{j!(i-j)!} \times \frac{2^d}{2^i}$$

Since UC has 2^d equally probable output ports, then the probability an abitrary output port (UC or LC) to transmit a cell is $(B_{case-1}(p) + B_{case-2}(p) + B_{case-3}(p))2^{-d}$ which is the passthrough of B_{DSW}.

References

[1] 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.

[2] M. Kawarasaki and B. Jabbari. B-ISDN architecture and protocol. *IEEE J. Selected Areas in Communications*, 9(9):1405–1415, Dec. 1991.

[3] D. Delisle and L. Pelamourgues. B-ISDN and how it works. *IEEE Spectrum*, 28(8):39–42, Aug. 1991.

[4] Martin de Prycker. *Asynchronous Transfer Mode - solution for broadband ISDN.* Ellis Horwood, 1991.

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

[6] Reza Rooholamini, Vladimir Cherkassky, and Mark Garver. Finding the right ATM switch for the market. *IEEE Computer*, 27(4):17–28, Apr. 1994.

[7] A. Saha and M. D. Wagh. Performance analysis of banyan networks based on buffers of various sizes. *IEEE Proc. INFOCOM'90*, 1:157–164, 1990.

[8] K. E. Batcher. Sorting networks and their applications. *AFIPS Proc. 1968 Spring Joint Computer Conf.*, 32:307–314, 1968.

[9] J. S. Turner. New directions in communications (or which way to the information age?). *IEEE Commun. Mag.*, 24(10):8–15, Oct. 1986.

[10] Fouad A. Tobagi, Timothy Kwok, and Fabio M. Chiussi. Architecture, performance, and implementation of the Tandem Banyan fast packet switch. *IEEE J. Selected Areas in Communications*, 9(8):1173–1193, Oct. 1991.

[11] Toshihiro Hanawa et al. Multistage interconnection networks with multiple outlets. *1994 International Conference on Parallel Processing*, I:1–8, 1994.

[12] M. Al-Mouhamed, H. Yousef, and W. Hassan. A Parallel-Tree Switch Architecture for ATM networks. *Inter. Journal of Communication Systems*, Vol 11, No 1, 1997.

[13] J. T. schwartz. The Burroughs FMP machine. *Ultracomputer note 5, Courant Institute, New York University*, 1980.

[14] M. Kumar and J. R. Jump. Performance of unbuffered shuffle-exchange networks. *IEEE Trans. on Computers*, C-35(6):573–578, June 1986.

[15] T. T. Lee and S. C. Lieu. Broadband packet switches based on dilated interconnection networks. *IEEE Trans. on Communications*, Vol 42(2/3/4):732–744, 1994.

[16] Mark W. Garrett. A service architecture for ATM: from applications to scheduling. *IEEE Network*, 10(3):6–14, May/June 1996.

[17] G. D. Stamoulis, M. E. Anagnostou, and A. D. Georgantas. Traffic source models for ATM networks: a survey. *Computer Communications, Butterworth-Heinemann Ltd*, 17(6):428–438, Jun. 1994.