A 3-STAGE INTERCONNECTION STRUCTURE FOR VERY LARGE PACKET SWITCHES

Soung C. Liew and Kevin W. Lu

Bell Communications Research
445 South Street
Morristown, NJ 07960-1910, U.S.A.

Abstract

This paper proposes a 3-stage broadband packet switch architecture with more than 16,000 ports for a future central office. The switch is constructed by interconnecting many independent switch modules of small size which can be implemented using modifications of various well-studied switch fabric designs. Multiple paths are provided for each input-output pair, and the technique of channel grouping is used to decrease delay and increase throughput. A datagram packet routing approach is adopted in order to eliminate table lookup that will be required by virtual-circuit routing. Ways of guaranteeing the sequence integrity of packets are discussed. We estimate from our performance analyses that switches with acceptable performance of 32,768 ports can be constructed based on switch fabrics of no more than 128 ports.

I. Introduction

"Divide-and-Conquer" is a popular engineering technique for designing systems of large scale. A good approach to designing a very large switching system, for example, is to interconnect many independent small switch modules in a way that satisfies the overall switching requirements. Most studies of packet-switch implementations [1], [2], switch control [1], [3], [4], performance analysis [3],[5], and prototypes [1], [6] have been restricted to small switch fabrics that do not scale easily. It is now time to address the challenge of building a very large packet switch based on small switch fabrics.

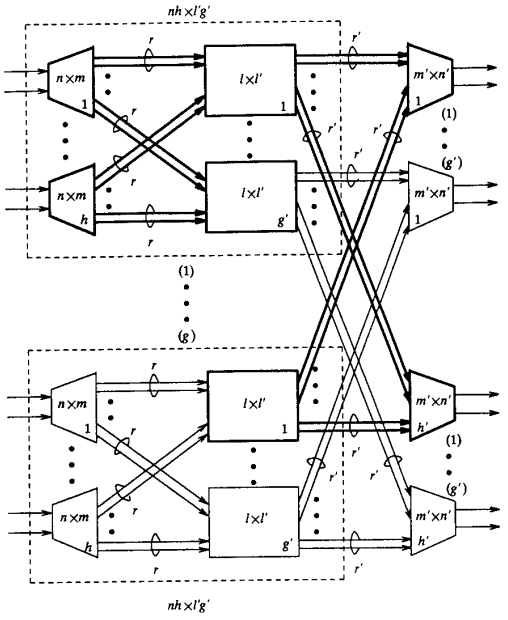
Toward this end, the present paper proposes a packet switch architecture which consists of three stages of small switch modules, each of which in turn can be realized by several switch fabrics. Several points of interest regarding this work are presented below.

- The proposed switch structure has a modular design that scales easily to dimensions larger than 16,000 ports × 16,000 ports. Thus, a packet switch with Terabit capacity can be achieved with a per-port transmission capacity of only 150 to 200 Mb/s. It is difficult to realize a Batcher-banyan switch [1], [6] of this scale because of stringent synchronization requirements for the self-routing elements at each stage of the switch. The modular growth of the Knockout switch described in [2] will encounter significant difficulties for 16,000 ports also, since each bus in the architecture must then have a fanout of more than 16,000.
- Lee has proposed a 2-stage nonblocking modular switch [7] in which the interconnection complexity grows faster than linearly with the overall switch size. For a $128^2 \times 128^2$ 2-stage modular switch based on switch fabrics of 128 input ports, the number of interconnections is $128^3 = 2,097,152$. While it is interesting and challenging to explore the possibility of building a very large nonblocking switch, the nonblocking property may be too stringent from an engineering viewpoint. By abandoning the nonblocking property, in our 3-stage design, we can achieve a switch of similar size with only 131,072 interconnections while maintaining acceptable performance.
- The technique of channel grouping (i.e., providing more than one physical output port for each physical destination address) [8] is used to improve the performance of the individual switch modules. This paper presents several switch module designs, and shows that for a given switch size, a switch module with channel grouping is simpler to realize than one without channel grouping. For instance, a Batcher-banyan switch module with channel grouping can be realized simply by truncating the last few stages of the original Batcher-banyan network.
- There are multiple paths between any input and any output in our overall switch architecture. We assume datagram routing in which packets of a given service session may travel over different paths within the switch architecture. This means packets may arrive at the output out-of-sequence, due to delay differences of the paths. However, this paper shows that sequence integrity of packets can be maintained automatically by proper design of the individual switch modules. The same technique can be extended to other multistage switches with channel grouping.

II. The General 3-Stage Switch Architecture

The general switch architecture we propose is illustrated in Fig. 1. The dimensions of the first-stage, second-stage, and third-stage switch modules are $n \times m$ ($m \geq n$), $l \times l'$, and $m' \times n'$ ($m' \geq n'$), respectively. The switch modules are nonblocking internally. For interconnection of the first-stage and second-stage modules, the modules are divided into g partitions called the *first-stage-second-stage partitions*. The figure shows the 1st and the gth enclosed in dashed boxes. There are no interconnections between partitions, but within each partition, each first-stage module is connected to each second-stage module via r links.

CH2829-0/90/0000-0771 $1.00 © 1990 IEEE



$$N = n\,g\,h = n'g'h'$$
$$g' = \frac{m}{r} \quad g = \frac{m'}{r'}$$
$$h = \frac{l}{r} \quad h' = \frac{l'}{r'}$$

Figure 1: General structure of a 3-stage switch.

We shall refer to the links that interconnect two switch modules as a *channel group*, and the number of links within each channel group as the *channel group size*. In each partition, there are h first-stage modules, g' second-stage modules, and hg' channel groups of size r.

A similar, but not identical, partitioning strategy is used for interconnection of the second-stage and third-stage modules. Specifically, the third-stage modules are divided into g' partitions. One second-stage module out of each first-stage-second-stage partition is chosen to be included in each of the g' second-stage-third-stage partitions. As before, there are no interconnections between partitions, but each second-stage module is connected to each third-stage module within the same partition via a channel group of size r'. In Fig. 1, two second-stage-third-stage partitions are shown. The darkened lines show the interconnections of one partition, while the undarkened lines show the interconnections of the other. In each partition, there are g second-stage modules, h' third-stage modules, and $h'g$ channel groups of size r'. Note that although there is more than one path between a first-stage module and a third-stage module, these paths belong to the same *path group* consisting of two channel groups.

It is interesting to note that when $l = l' = 1$ (i.e., the second-stage switch modules are removed), $r = r' = 1$, $h = h' = 1$, $m = g'$, $m' = g$, and $n' = 1$, the 3-stage architecture reduces to Lee's 2-stage architecture [7]. When $n = n' = m = m' = l = l'$ (i.e., switch modules of all three stages are of the same size), switch performance will be degraded at the first and second stages under high input loads because of congestion on the interconnections. By having $m > n$ and $m' > n'$, the loading on the interconnections can be reduced, and therefore the throughput of the switch can be increased.

It is not necessary to have $r > n$ or $r' > n'$ because no more than n or n' packets will travel simultaneously through a first-stage or third-stage switch module, respectively. With $r = n$ and $r' = n'$, there is always enough capacity to carry the packets to their destinations, but there will be a very large number of interconnections (N^2/l). With $r < n$ or $r' < n'$, the switch is blocking because there may not be enough capacity to carry the traffic between two modules. However, if the switch is properly designed, the likelihood of this event can be made extremely small.

The rest of this paper will consider only the symmetric case in which $n = n'$, $m = m'$, $l = l'$, $g = g'$, $h = h'$, and $r = r'$. The total number of switch ports is

$$N = ngh. \tag{1}$$

A total of $\log_2 N$ bits is necessary for routing purposes, and of these, $\log_2 g$ bits are used for routing at the first stage, $\log_2 h$ bits at the second stage, and $\log_2 n$ at the third stage. The total number of interconnections is

$$I = 2mgh. \tag{2}$$

From the above, we get

$$\frac{I}{N} = \frac{2m}{n}. \tag{3}$$

We shall call m/n the *expansion ratio*, referring to the fact that this is the ratio of the number of intermediate links in the switch architecture to the number of inputs or outputs. Equation (3) shows that the only way to increase N without increasing I is to decrease the expansion ratio m/n, but this will result in degradation in performance. Thus, we see that although an optimal switch should have large N, small I, and good performance, these objectives conflict with each other. Our task, therefore, is to design the overall switch to achieve the best compromise between these objectives.

Global FIFO Property

Because of channel grouping, there are r possible paths in our switch architecture over which a packet can travel from input to output. It is not difficult to envision situations in which two packets are reversed in order at the output because they have traveled over different paths. On the other hand, it is easier to achieve the first-in-first-out (FIFO) in our switch structure than in a switch structure with more than one intermediate switch module interconnecting a first-stage switch module and a third-stage switch module. In such a case, packets of an input-output pair may travel through different intermediate switch modules at different times, and sequence integrity is difficult to maintain unless the intermediate modules are somehow coordinated and synchronized. For our switch architecture, on the other hand, we only need to focus on one path group to tackle the FIFO problem, thus allowing the intermediate switch modules to function independently.

Whether the proposed 3-stage switch architecture is FIFO depends on the buffering strategies used by switch modules at the

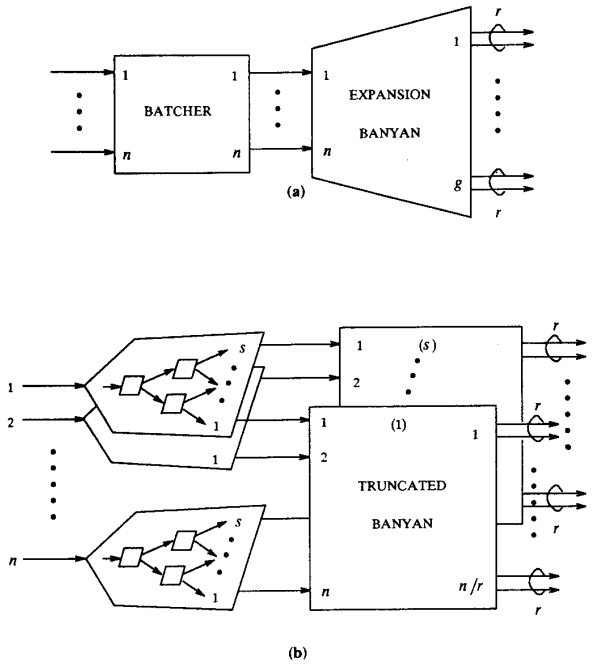


Figure 2: (a) Batcher-banyan implementation of a first-stage input-buffered switch module, (b) Expansion network.

different stages, as well as on the switch designs themselves. This paper considers three different buffering strategies: input queueing, output queueing, and packet dropping [5]. With input queueing, an arriving packet enters a FIFO buffer on its input and waits for its turn to access its destination output. When there is channel grouping on the output, the packet may be routed to any output of the destination output group. The channel group size is the maximum number of packets that can be cleared from the same output address simultaneously. With output queueing, a FIFO buffer is allocated to each output group, and arriving packets destined for this output group are immediately placed in this buffer. Again, packets may be routed to any output of their destination output group. With packet-dropping, there is no queueing at input or output ports. Any packets not cleared in one time slot are simply dropped from the system.

For a structure with input queueing at the first stage, packet dropping at the second, and output queueing at the third, it is not difficult to see that sequence integrity of undropped packets is maintained, regardless of the design details of the switch modules. In fact, it can be shown that in general at least one the three stages must be packet-dropping in order to maintain FIFO. However, with the particular output-queueing switch modules proposed in the next section, the first-stage modules and/or the second-stage modules can also be output queueing without violating FIFO. Specifically, the proposed output-queueing switch modules assume an implicit time relationship between packets of the same channel group that arrive in the same time slot, and this implicit time relationship is automatically preserved even when multiple stages of these output-queueing switch modules are cascaded.

III. Switch Module Designs

We now consider the internal structures of the switch modules. For this paper, the switch hierarchy is ordered as: *very large switch* → *switch module* → *switch fabric* → *switch element*. This section addresses the lowest three levels of this hierarchy. The design of switch modules is basically similar to the design of small switches, and as such, results and insights gained from other studies [1], [2], [7] can be drawn upon to design the switch modules. We show in the rest of this section ways in which channel grouping allows us to simplify switch designs.

Input-buffered Switch Module with Channel Grouping

Only input-buffered switch modules at the first stage will be considered here. Two possibilities are discussed below.

(i) *Modified Batcher-Banyan Network:* The structure of the first design is basically a modification based on the expansion Batcher-banyan network described in [7]. The Batcher network [10] sorts the packets according to their destination addresses, either in ascending order or descending order, and the expansion banyan network then routes the packets to their destinations. It is well known that this switch structure is nonblocking [1], [7], [11].

The expansion banyan network is basically a $gr \times gr$ banyan network with some of the 2×2 switch elements omitted [7]. A schematic of the expansion banyan network is illustrated in Fig. 2(b), where $ns = gr = m$, and n, s, g and r are all multiples of 2. By using a tree-branching interconnection of switch cells, each output from the Batcher network is expanded into s lines. Then, s $n \times n$ banyan networks are used to further route the packets. With a control scheme that makes sure that no more than r packets destined for the same output group enter the Batcher network in the same time slot (e.g., modifications based on [3],[4]), the network is nonblocking if $\log_2 gr$ bits are used for routing [7]. To route packets in the banyan network, we could generate another $\log_2 r$ address bits (in addition to $\log_2 g$ bits of original output-group address) in such a way that packets originally having the same output address are routed to different output ports of the same output group.

Because the output lines of a given output group are indistinguishable to packets, the above scheme is more complex than is actually needed. The following can be shown:

- A truncated $n \times gr$ banyan network with the last $\log_2 r$ stages removed is nonblocking if the input packets are presorted and at most r packets are destined for each of the g output groups.

With this simplification, only the $\log_2 g$ original output address bits are required for routing.

(ii) *Expansion-Concentration Network:* The second design consists of an expansion stage followed by a concentration stage (see Fig. 3(a)). Unlike the Batcher network of the previous design in which all $\log_2 g$ bits of the destination address are examined in each sorting element, all switch cells in this design are similar to those in a banyan network in that only one address bit needs to be examined.

Each input port has an associated expansion network, and packets are routed according to their destination output groups.

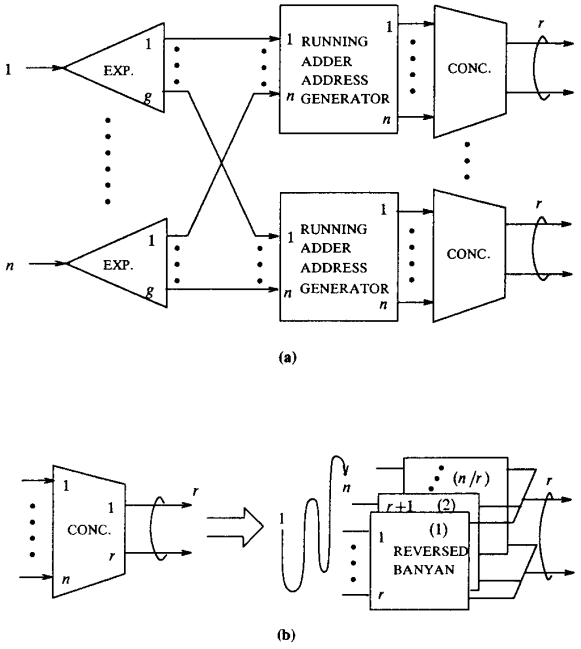


Figure 3: (a) Expansion-concentration implementation of a first-stage input-buffered switch module, (b) Implementation of a concentrator.

Note, in particular, that the number of output ports for each expansion network is g, the number of output groups for the switch module, rather than gr, the number of output ports for the switch module. This reduces the fanout by a factor of r. Packets destined for the same output group are then collected and fed into a running-adder address generator [1], [12], before they are cleared from the switch module at the r output ports of a concentrator. The address generator produces a set of adjacent destination addresses in the concentrator for the active packets. One possible addressing scheme is to give the first active packet (from top to bottom) the first output address, the second active packet the second output address, and so on. If there is an activity bit for each packet, then the address assigned to a packet is the sum of all activity bits belonging to the packets above it.

The structure of a concentrator, as shown in Fig. 3(b), consists basically of an interconnection of several reversed banyan networks. A reversed banyan network is a mirror image of a regular banyan network in which the inputs and outputs are reversed. The n inputs to the concentrator are partitioned into n/r groups of r inputs each, and each group is fed into an $r \times r$ reversed banyan network. The corresponding outputs of the reversed banyan networks (from top to bottom) are connected to common buses, resulting in exactly r output ports. Given at most r packets to each output group, the concentrator is nonblocking if the routing at the kth stage is done according to the kth least significant bit (as opposed to the kth most significant bit in the regular banyan network) [11], [12].

The address-generation scheme discussed above is top-to-bottom, and when there are fewer than r packets, the packets tend to concentrate on the upper portion of the outputs. If we want to distribute the packets more evenly over the outputs (e.g., when the switch modules at the next stage are input-buffered), the address-generation scheme above can be easily modified by adding an offset quantity a, $0 \leq a \leq r-1$, to the sum of the activity bits of the upper inputs. That is, if S is sum of the activity bits of the upper inputs, the generated output address will be $a+S \pmod r$. It can be shown (using Theorem 5 and Theorem 13 in [11]) that the same routing scheme is still nonblocking. The a value can then be varied from time slot to time slot in order to distribute the packets more evenly among the outputs.

The expansion-concentration network just described eliminates the sorting requirement of a Batcher-banyan switch design. This is accomplished, however, through the addition of address generators. As in the Batcher-banyan design, a control scheme is required to make sure no more than r packets for the same output group enter the switch simultaneously.

Input-buffered switch modules for the second and third stages can be implemented based on modifications of the above designs. Because of channel grouping on the input ports, however, additional control mechanisms will be required to maintain the FIFO property of the modules. These FIFO-guaranteeing mechanisms will not be discussed in this paper.

Packet Dropping

Packet-dropping switch modules can be designed as above, except that buffers at the inputs are omitted. For illustration, consider a packet-dropping switch module at the second stage with dimensions $hr \times hr$. If we use the Batcher-banyan design described earlier in this section, then $n \rightarrow hr$, $g \rightarrow h$ and $s \rightarrow 1$. In particular, the banyan network is $hr \times hr$ and not expanded, but as before, the last $\log_2 r$ stages of the banyan network can be omitted.

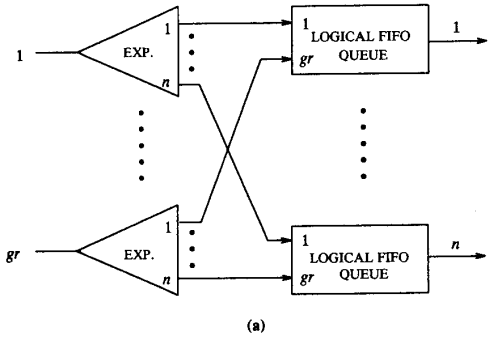
Output Queueing

Output-buffered switch modules can be implemented based on modifications of the second design of the input-buffer switch modules. We first consider an output-buffered switch module at the third stage.

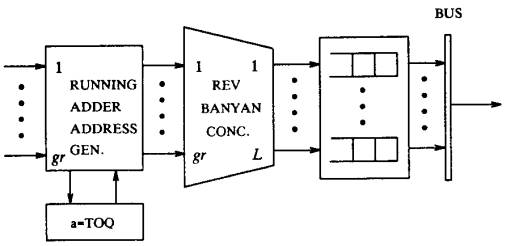
Figure 4(a) shows the general structure of the switch module. There are gr expansion networks, one for each input. The $\log_2 n$ address bits of a packet are used to route the packet to one output of the expansion network. The corresponding outputs of the expansion networks are collected and fed to the logical FIFO output queues, which simply clear the buffered packets on a first-in-first-out basis.

To implement a logical FIFO output queue, an approach like the one used in the Knockout Switch [2] can be adopted, in which a concentrator, shifter, and multiple packet buffers are used. In particular, the concentrator is used to reduce the number of inputs that need to be buffered simultaneously. Here, we follow the same general approach, but propose a different structure to replace the concentrator and shifter, which yields a smaller count of switch elements.

A schematic diagram for the FIFO output queue is shown in Fig. 4(b). The n input lines are concentrated to L lines by a reversed-banyan concentrator. At any given time slot, a packet is selected from one of the L buffers and transmitted to the output over a common bus. To maintain the FIFO property, packets



(a)



(b)

Figure 4: (a) Implementation of a third-stage output-buffered switch module, (b) Implementation of a logical FIFO queue at the third stage.

at the buffers exit the switch in a round-robin fashion, proceeding from top to bottom and wrapping back. Packets also enter the buffers in round-robin fashion, from top to bottom, so that packets from earlier time slots always precede packets from later time slots. To ensure this, a running-adder address generator is cascaded with a reverse banyan network. The running-adder address generator assigns each packet an output address which is equal to $a + S \pmod{L}$, where $0 \leq a, S \leq r - 1$. Here, a is set to the tail-of-queue (TOQ), i.e., the buffer just below the one entered by the last packet in the previous time slot. S is the total number of active packets at inputs above the input of the packet concerned. The quantity a in the next time slot is simply the sum of a and the number of packets in the current time slot. The number of routing switch cells required for each reversed-banyan concentrator is the sum of those in the gr/L $L \times L$ banyan networks (i.e., $n \rightarrow gr$, $r \rightarrow L$ in Fig. 3(b)). This is $(gr/2)\log_2 L$, which is of smaller order than grL in the Knockout Switch [2].

Under conditions of homogeneous traffic, it is unlikely (with probability $< 10^{-6}$) that there will be more than 8 new incoming packets for the same output in a given time slot, regardless of gr [2] . In practice, we can take into account an occasional surge in instantaneous traffic by fixing L at 16. The buffers do not have to be very deep, either. Reference [2] shows that it is sufficient to have enough storage for 40 packets per output. Thus, for $L = 16$, each buffer needs to be only 3 packets deep. In general,

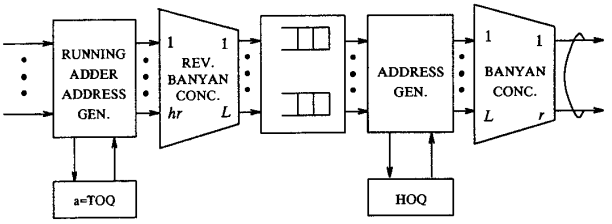


Figure 5: Implementation of a logical FIFO queue at the second stage.

the buffering requirement in output-buffered switch designs is no worse than that in input-buffered switch design.

For output-queueing switch modules at the first stage, $gr \rightarrow n$ and $n \rightarrow g$ in Fig. 4(a). Similarly, for output-queueing switch modules at the second stage, $gr \rightarrow hr$ and $n \rightarrow h$ in the figure. In either case, there is channel grouping on the outputs, so the logical FIFO queue structure in Fig. 4(b) must be modified. Suppose the channel group size is given by $r = L/2^i$ for some $i \in \{0, 1, 2, \ldots\}$, and suppose we want to preserve the order of packets in the logical FIFO queue by transmitting the first packet on the top output, the second packet on the next output, and so on. The incentive for this is that the switch module at the next stage will then know the implicit sequence of the simultaneously received packets, and it will put the packets in the FIFO output buffers according to their implicit sequence. In this way, the sequence integrity of packets is preserved even when multiple stages of output-buffered switch modules are cascaded together.

Figure 5 illustrates a way of achieving this at the second stage by cascading an address generator and a banyan concentrator after the buffers. The banyan concentrator is similar to that in Fig. 3(b), except here we have 2^i $r \times r$ banyan networks connected by common buses. Furthermore, both regular and reversed banyan networks can be used here. The head-of-queue (HOQ) is the input containing the first packet of the logical FIFO queue. To map the first r packets to the r outputs of the next concentrator, the address generator performs the following mapping: output address = input address $- HOQ \pmod{L}$. Only packets with output addresses in the range of 0 to $r - 1$ will be transmitted through the concentrator. The HOQ in the next time slot is simply $HOQ + \max(S, r) \pmod{L}$, where S is the sum of the activity bits of the L head-of-line packets in the L buffers. For the desired cyclic mapping, this banyan concentrator is nonblocking [11].

The reversed banyan networks in Fig. 4(b) and Fig. 5 may be difficult to implement when gr and hr are large. Figure 6 shows a decomposition method for solving this problem, using a third-stage module as an example.

IV. Switch Performance

We now consider the performance of the 3-stage switch architecture in order to derive reasonable ranges for the various switch parameters. Only the results of our analyses and simulations will be presented here.

To determine the merits of the 3-stage switch, we consider the number of interconnections between modules, and performance

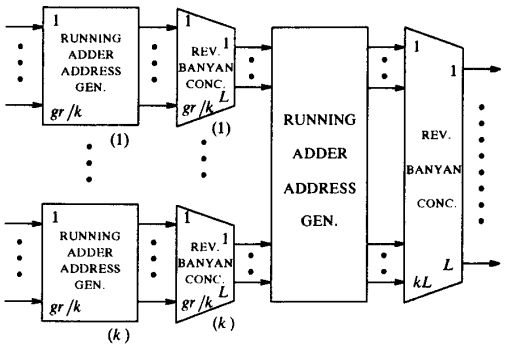


Figure 6: Decomposition of a logical FIFO queue at the third stage

measures such as the maximum throughput, packet loss probability, and mean delay. The rest of this section focuses on a specific design that uses input queueing, packet dropping, and output queueing for the first, second and third stages, respectively. Each of the switch modules is internally nonblocking. In our analyses, we assume homogeneous traffic. Packets arrive at input ports with fixed probability for each time slot. They have equal probability of being addressed to any output. Furthermore, there is no correlation between packets arriving in different time slots or at different inputs.

We first consider the maximum throughput for input queueing at the first stage. It is well known that large input-buffered switch has a maximum throughput of 0.586 per line [5]. We have modified the analysis in [5] to take into account the asymmetry and channel grouping in our switch modules. That is, in our first-stage switch module, the number of outputs (m) is different from the number of inputs (n), and a packet needs to access only one of r outputs of its target output group. Tables 1(a) and 1(b) present the maximum throughputs per input for various values of g/n, m/n, and r in the limit $n \to \infty$. The limit is taken to simplify the analyses, but the maximum throughputs thus found provide a very tight upper bound for cases with finite but large n (e.g., $n = 128$). This has been confirmed by computer simulation of the input-buffered switch modules. Table 1(a) clearly shows that the technique of channel grouping (i.e., increasing r while fixing g/n) can increase the maximum throughput substantially.

Recall now that it is desirable to minimize the quantity $I/N = 2m/n$ in order to decrease I or increase N. On the other hand, as shown in Table 1(b), decreasing the expansion ratio m/n beyond a certain limit results in a maximum throughput of less than 1. Thus, the ideal value for m/n is just above this limit. For $r = 8$, for example, the table shows that the ideal value for m/n is 4.

Table 2 shows the mean delays (in terms of number of time slots) at the first and third stages, and packet dropping probability at the second stage, for various switch parameters with $m/n = 4$, and $r = 16$. An offered load of $p = 0.95$ per input is assumed, and only the sets of parameters that yield acceptable performance are shown. The mean delay at the first stage is determined by simulations. The expansion ratio m/n is large enough that most packets get through the first stage without

Table 1: Maximum throughput of an input queue at the first stage with (a) g/n kept constant while both g and $n \to \infty$ and (b) $m/n(= gr/n)$ kept constant while both m and $n \to \infty$.

(a)

r	$\frac{g}{n}$ = $\frac{1}{32}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{2}$	1	2	4	8	16	32
1	0.031	0.061	0.117	0.219	0.382	0.586	0.764	0.877	0.938	0.969	0.984
2	0.061	0.121	0.233	0.426	0.686	0.885	0.966	0.991	0.998	0.999	1.000
4	0.123	0.241	0.457	0.768	0.959	0.996	1.000	1.000	1.000	1.000	
8	0.245	0.476	0.831	0.991	1.000	1.000					
16	0.487	0.878	0.999	1.000							
32	0.912	1.000	1.000								

(b)

r	$\frac{m}{n}$ = 1	2	4	8	16	32
1	0.586	0.764	0.877	0.938	0.969	0.984
2	0.686	0.885	0.966	0.991	0.998	0.999
4	0.768	0.959	0.996	1.000	1.000	1.000
8	0.831	0.991	1.000			
16	0.878	0.999				
32	0.912	1.000				
64	0.937					
128	0.955					
256	0.968					
512	0.978					
1024	0.984					

Table 2: Comparison of switch performance for various switch designs.

N	n	m	l	r	g	h	I	$D_1^{(I)}$	P_2	$D_3^{(O)}$
16,384	32	128	1024	16	8	64	131,072	1.0	0.17×10^{-6}	10.43
16,384	64	256	256	16	16	16	131,072	1.0	0.13×10^{-6}	10.46
16,384	128	512	64	16	32	4	131,072	1.0	0.41×10^{-7}	10.48
32,768	64	256	512	16	16	32	262,144	1.0	0.16×10^{-6}	10.46
32,768	128	512	128	16	32	8	262,144	1.0	0.93×10^{-7}	10.48
65,536	64	256	1024	16	16	64	524,288	1.0	0.17×10^{-6}	10.46
65,536	128	512	256	16	32	16	524,288	1.0	0.13×10^{-6}	10.48
65,536	256	1024	64	16	64	4	524,288	1.0	0.41×10^{-7}	10.49
131,072	128	512	512	16	32	32	1,048,576	1.0	0.16×10^{-6}	10.48
131,072	256	1024	128	16	64	8	1,048,576	1.0	0.93×10^{-7}	10.49

I Number of interconnections,
$D_1^{(I)}$ Mean delay of input queueing at the first stage,
P_2 Packet loss probability of packet dropping at the second stage,
$D_3^{(O)}$ Mean delay of output queueing at the third stage.

queueing. At the second stage, the mean delay is always one time slot, since there is no queueing. The packet dropping probabilities are less than 10^{-6} for the parameters concerned. The mean delay at the third stage depends only on m and the offered load p, and the dependence on m is weak for large m (cases of interest to us). For $p = 0.95$, the third-stage mean delay time is about 10.5 time slots. The table shows that switches with acceptable performance of up tp 131,072 ports can be built using switch modules of no more than 512 ports (i.e., $n, m, l \leq 512$). Recall that each module is in turn made up of several switch fabrics. For the last entry in the table, $n = 256$, $m = 1024$, and $l = 128$. Using the switch module designs proposed in the previous section, the size the largest switch fabric in the switch modules of all stages is no more than 256.

It is interesting to note that the magnitudes of the parameters in Tables 1 and 2 are mainly determined by the need for a small packet dropping probability at the second stage, since there is virtually no waiting at the first stage and the delay at the third stage is largely independent of the switch parameters. The configuration of input buffering at the first stage, packet dropping at the second, and output buffering at the third, is chosen for analysis here because it maintains the sequence integrity of packets, and because it provides a good example to illustrate the three queueing strategies. In practice, we could also replace the packet dropping switch modules at the second stage with output-buffered switch modules without sacrificing the sequence integrity, provided the output-buffer modules are implemented as in the previous section. The advantage of this is that the expansion ratio m/n, and hence the number of interconnections $I = 2Nm/n$, can be further reduced.

V. Conclusions

We have proposed a class of 3-stage architectures for building a very large packet switch using small switch modules. Datagram routing has been adopted to avoid internal path-hunting for service setup and table-lookup for routing packets. We have shown that the FIFO property can be preserved in our overall switch architecture by designing the individual switch modules properly, even though there are multiple paths between any input-output pair in the architecture. Specifically, we have proposed channel-grouping output-buffered switch modules which preserve the FIFO property when cascaded. In addition, this paper has presented ways in which various well-studied switch fabric designs can be simplified for our switch modules.

Two distinct differences between our performance analysis and other studies is that our switch modules are dimensionally asymmetric and that we use channel grouping (i.e., we give more than one output ports to each output address). A table listing the maximum throughputs for input-buffered switch modules has been provided, and the use of channel grouping to increase throughput established.

Based on a structure with input queueing at the first stage, packet dropping at the second stage, and output queueing at the third stage, we conclude that 3-stage packet switches with a very large number of ports and reasonable interconnection complexity can be achieved. In particular, switches with acceptable performance of up to 131,072 ports can be constructed based on switch fabrics with no more than 256 ports. As an example, we estimate that a 3-stage switch with 32,768 ports can be built using switch modules of no more than 512 ports, with each switch module in turn being built using switch fabrics of no more than 128 ports. The corresponding number of interconnection channels between the switch modules is 262,144. The number of interconnections can be further reduced if the packet-dropping switch modules are replaced with output-buffered switch modules. In either case, with a capacity of 155 Mb/s per port (corresponding to the SONET STS-3 signal), the total capacity of the switch is about 5 Tb/s. Switch fabrics of 128 ports and 155 Mb/s are certainly within the reach of the current VLSI technologies.

Acknowledgements

We thank Tony Lee for generously sharing his knowledge and expertise with us. Howard Lemberg's comments have significantly improved this paper.

References

[1] A. Huang and S. Knauer, "Starlite: A Wideband Digital Switch," *Globecom 84 Proc.*, pp. 121–125.

[2] Y. Yeh, M. Hluchyj, A. Acampora, "The Knockout Switch: A Simple Modular Architecture for High-Performance Packet Switching," *IEEE J. on Selected Areas in Commun.*, vol. SAC-5, no. 8, pp. 1274–83, Oct. 1987.

[3] J. Hui and E. Arthurs, "A Broadband Packet Switch for Integrated Transport," *IEEE J. on Selected Areas in Commun.*, vol. SAC-5, no. 8, pp. 1264–73. Oct. 1987.

[4] B. Bingham and H. Bussey, "Reservation-Based Contention Resolution Mechanism for Batcher-Banyan Packet Switches," *Elect. Lett.*, vol. 24, no. 13, pp. 772-773, June 1988.

[5] M. Hluchyj and M. Karol, "Queueing in High-Performance Packet Switching," *IEEE J. on Selected Areas in Commun.*, vol. 6, no. 9, pp. 1587–97, Dec. 1988.

[6] M. Littlewood, "Sunshine: A Broadband Packet Switch," Submitted to *International Switching Symposium*, 1990.

[7] T. Lee, "A Modular Architecture for Very Large Packet Switches," to appear in *Globecom '89 Proc.*.

[8] A. Pattavina, "Multichannel Bandwidth Allocation in a Broadband Packet Switch," *IEEE J. on Selected Areas in Commun.*, vol. 6, no. 9, pp. 1489–99, Dec. 1988.

[9] H. Suzuki *et al.*, "Output-Buffer Switch Architecture for Asynchronous Transfer Mode," *ICC '89 Proc.*, vol. 1, pp. 99–103.

[10] K. Batcher, "Sorting Networks and Their Applications," *AFIPS Proc. Spring Joint Comput. Conf.*, 1968, pp. 307–314.

[11] D. Lawrie, " Access and Alignment of Data in an Array Processor," *IEEE Trans. on Comput.*, vol. c-24, no. 12, pp. 1145–55, Dec. 1975. *IEEE J. on Selected Areas in Commun.*, vol. SAC-6, no. 9, pp. 1455–67, Dec. 1988.

[12] T. Lee, "Nonblocking Copy Networks for Multicast Packet Switching," *IEEE J. on Selected Areas in Commun.*, vol. SAC-6, no. 9, pp. 1455–67, Dec. 1988.