Chapter 3

Fast Packet Switch Architecture

Having established the basic concept of the fast packet switch and given an impression of the context into which it fits as a switching mechanism this chapter attempts to explore the architecture of the fast packet switch. A simple classification of fast packet switch designs is first introduced. A number of fast packet switch designs which have recently appeared in the literature are then reviewed and some comparisons drawn. Finally, from the existing literature, an elementary performance comparison between the three major classes of switch design is presented which will be developed in later chapters.

3.1 A Simple Classification of Switch Designs

Two fundamental components are required to construct a fast packet switch: switching and buffering; and the relative positioning of these components permits a simple classification of fast packet switch design, fig. 3.1. If the buffering remains external to the switch fabric the design is based upon a non-buffered switch fabric. Else, if the buffering is implemented within each of the switching elements forming the switch fabric a buffered switch fabric (or internally buffered) design results. Of the designs based upon a non-buffered switch fabric, if the buffering precedes the switch fabric the switch is classified as input buffered. Else, if the buffering follows the switch fabric the design is output buffered. An input buffered design requires much less hardware and fewer interconnections than a similar output buffered switch but its basic performance is only about half that of the ideal output buffered switch. This difference in performance results from an effect known as head of the line blocking [71, 76] which is discussed in the following section.

Input buffered switches may be classified according to whether the switch fabric is blocking or non-blocking. Blocking is said to occur when the transmission of an incident packet to a free output is temporarily prevented by other traffic within the switch fabric. A blocking switch will have a lower performance than an equivalent non-blocking fabric but will require fewer switching elements and interconnections. Various techniques are available to improve the basic performance of an input buffered

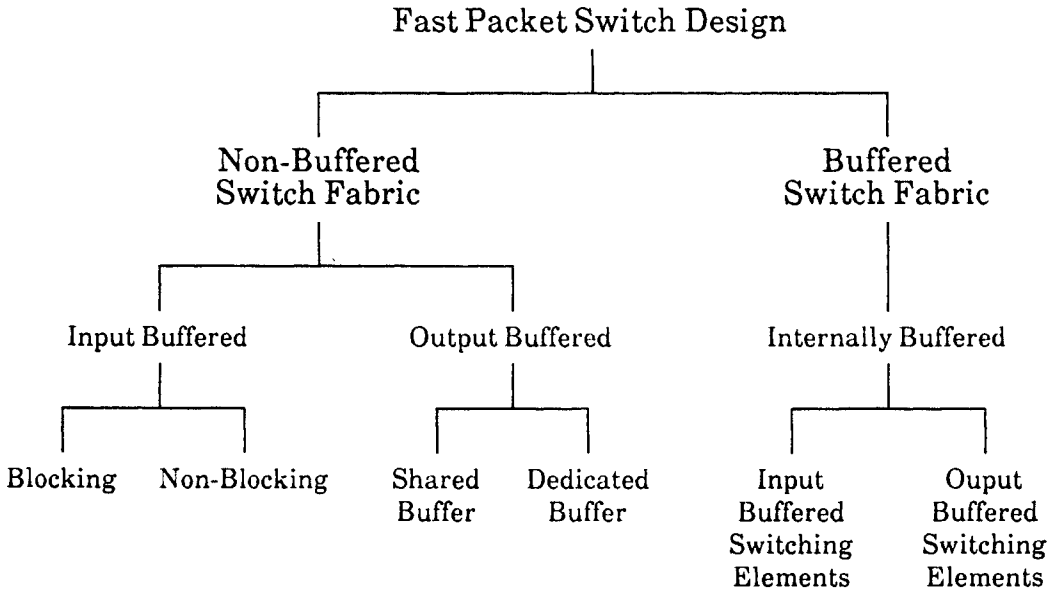

Figure 3.1: A simple classification of fast packet switch design.

switch towards that of the output buffered switch. The Cambridge Fast Packet Switch is an example of a blocking input buffered design while the three phase Batcher-banyan [71] is non-blocking.

Output buffered switches may share a pool of buffers between all output ports of the switch fabric else buffers may be dedicated to each output port. A shared design will require fewer buffers than a dedicated design. Starlite [70] provides an example of a shared buffer design while the Knockout switch [163] employs dedicated buffers.

In an internally buffered switch design each switching element within the switch fabric has its own buffers. These buffers may either be implemented on the input side of each switching element or upon the output side. Input buffered switching elements are much simpler to construct than are output buffered switching elements but they offer a lower performance. In general the performance of an internally buffered switch design using input buffered switching elements is similar to that of an input buffered switch design using a non-blocking switch fabric. Conversely, the performance of an internally buffered design with output buffered switching elements may approach that of an output buffered switch design for switching elements of large degree. Turner's switch [145, 148, 19] and the CSELT design [128, 56] offer examples of an internally buffered design using input buffered switching elements. Prelude [141, 31], the Bus Matrix Switch [120], the IBM Switch [4] and the TDM Bus design [36, 35] are all examples of designs based upon an output buffered switching element.

3.2 Input Buffered Switches

The banyan network is a multi-stage interconnection network employed as a switch fabric in both buffered and non-buffered designs. It is a blocking network but if all active incident packets are sorted into order based upon their destination output port number before being applied to the banyan network, non-blocking performance results [107, 88]. In a non-buffered switch fabric the Batcher sorting network may be used to provide this function and a non-blocking Batcher-banyan network results. This network requires that all packets be of the same length and that they be presented to the network in synchronism. It also requires any conflicts between packets contending for the same output port during the same timeslot to be resolved prior to the network. The Batcher-banyan switch fabric has been suggested for use both in input buffered and output buffered switch designs which will now be reviewed. A detailed discussion of the construction of Batcher, banyan and other interconnection networks will be presented in the following chapter.

The Three Phase Batcher-Banyan

The Batcher-banyan switch fabric will only offer non-blocking performance provided that no more than a single packet requests access to any switch output at the same time. The three phase Batcher-banyan switch [71] detects packets contending for the same output within the same timeslot by use of a three phase algorithm and causes requests that cannot be satisfied to be buffered at the input of the switch fabric. All packets are submitted to the switch fabric synchronously, and requests that cannot be satisfied in the current timeslot are re-submitted in the next. The basic structure of the switch is presented in fig. 3.2. Incoming packets are queued in the input buffers which act as first in first out (FIFO) queues. In phase I of the algorithm every port controller with a packet to transmit sends out a pilot packet which consists merely of the required switch output port number followed by the source input port number. These pilot packets are sorted into ascending order of output port number by the sorting network. This results in requests contending for the same output port emerging on adjacent ports of the sorting network. Every k^{th} output of the sorting network is fed back to every k^{th} port controller which allows the port controllers to decide which packets win the contention by comparing adjacent packets. At this stage, the input ports which originated the requests do not know the result of the arbitration, thus phase II of the algorithm, the acknowledgement phase, is necessary. In phase II every port controller that observed a successful pilot packet returns an acknowledgement to the originating port controller across the entire Batcher-banyan switch fabric. In phase III those input ports that receive an acknowledgement to their pilot request transmit the full packet across the switch fabric.

This fast packet switch design has two significant drawbacks, both of them a result of the size of the Batcher sorting network. A much larger number of switching elements are required to form a Batcher network than for a banyan network and although the switching elements of the Batcher network may be simpler to implement

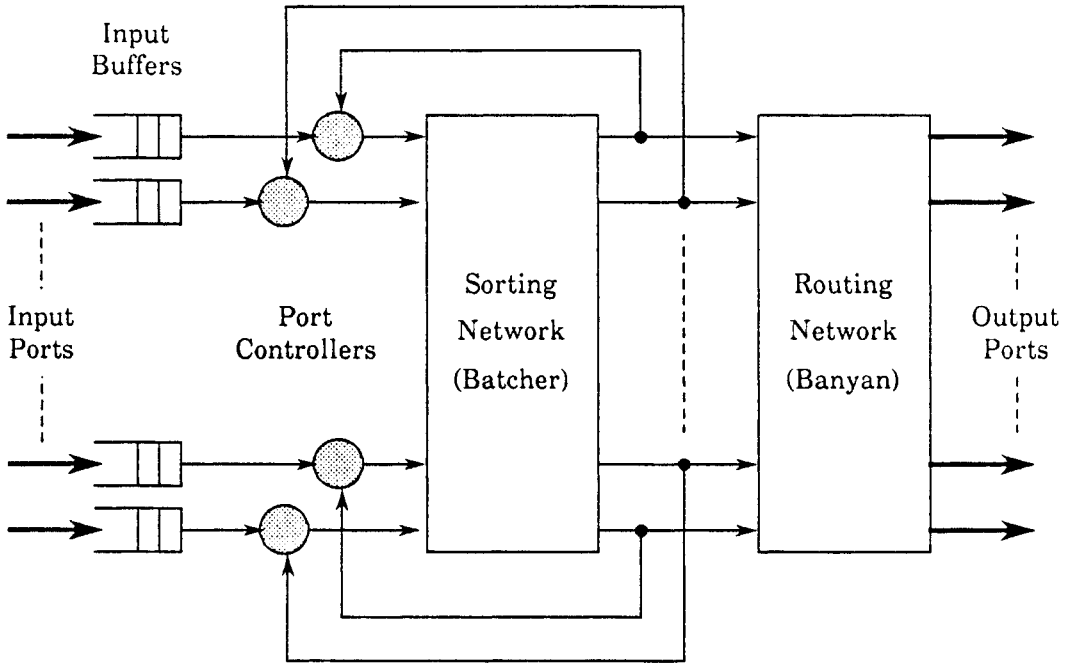

Figure 3.2: Basic structure of the three phase Batcher-banyan switch.

than those of the banyan, difficulties arise when partitioning the switch fabric for implementation in VLSI. One such implementation is discussed in [34] in which two VLSI chips have been designed for the switch fabric, one for the front end and one for the back end. In the 64×64 example switch discussed, the Batcher-banyan switch fabric has been partitioned into 9 stages, 7 of which are required for the Batcher sorting network and only two for the banyan routing network. Thus six stages of interconnection result from the Batcher network while only one is required for the banyan. Whilst an impressive example of layout and engineering permits the goal of a 256×256 Batcher-banyan switch fabric operating at 100 MHz with CMOS devices to look realistic, one is tempted to consider the performance of switch fabrics that suffer a small amount of blocking but are much smaller and simpler to partition and construct.

The second problem related to the size of the Batcher sorting network is that of the overhead incurred by the first two phases of the three phase algorithm. Phases I and II of the algorithm require a total of $\log_2 N(\log_2 N + 4)$ bit times to arbitrate between conflicting requests (where N is the size of the switch). The majority of this is due to delay through the Batcher sorting network. Thus for a large switch size, e.g. 1024×1024, a total of 140 bit times of overhead would be required for every packet. A popular packet size for use in the broadband ISDN is currently considered to be 128 bits, so such a switch would be operating at a basic efficiency of less than 50% at this packet size. A suggestion has been made to reduce the three phase algorithm to a two phase algorithm by the incorporation of two additional forms of

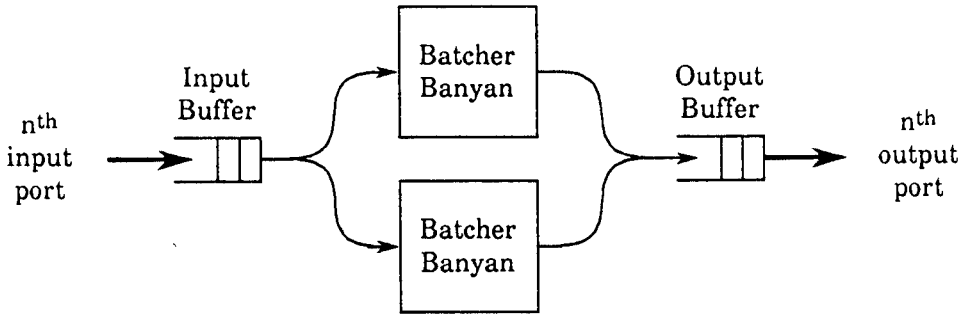

Figure 3.3: A two-plane switch structure.

interconnection network between the Batcher and banyan networks. This however considerably increases the size of an already large switch fabric.

Another problem, common to all designs of input buffered switch, is referred to as head of the line blocking. When a packet contends for an output port and loses, during a particular timeslot, it remains at the head of the FIFO queue at its input port until the following timeslot. It may be that there are other packets behind it on the queue destined for other output ports which could have been served during the current timeslot. These packets are blocked and this causes a reduction in throughput compared to output buffered and internally buffered switch designs. In input buffered switch designs that are asynchronous at the packet level, the transmission of packets other than the one at the head of the queue may be attempted on discovering that the first packet is blocked. This technique is often referred to as input queue by-pass. It improves the throughput of the switch and also reduces the sensitivity of the switch to the arrival statistics of the incident traffic. In a synchronous design, such as the above, this may only be achieved at the expense of repeating phases I and II of the algorithm on the second packet in the queue, after the arbitration of the first packet is completed, prior to phase III in every timeslot. This would significantly increase the overhead for every packet.

An alternative is to adopt a two-plane switch fabric as shown in fig. 3.3. The three phase algorithm of the second switch fabric is phased to commence just after the algorithm for the first switch fabric is completed so that packets blocked in the first fabric may be sent across the second. This requires each output port to be capable of handling two packets arriving at once. As there is no feedback from the output ports of the switch to the input ports, the output buffer must be dimensioned to reduce the probability of buffer overflow to acceptable proportions. Also the input port may be constructed to be capable of transmitting across both switch planes simultaneously, or alternatively only across a single plane at any one time. The most important aspect of adding the second switch plane is to provide redundancy for enhancing system reliability, but from the performance viewpoint it is interesting to question to what extent non-blocking operation is now required of the individual switch planes. Could the use of much simpler, blocking switch planes, in a two-plane structure achieve a similar performance?

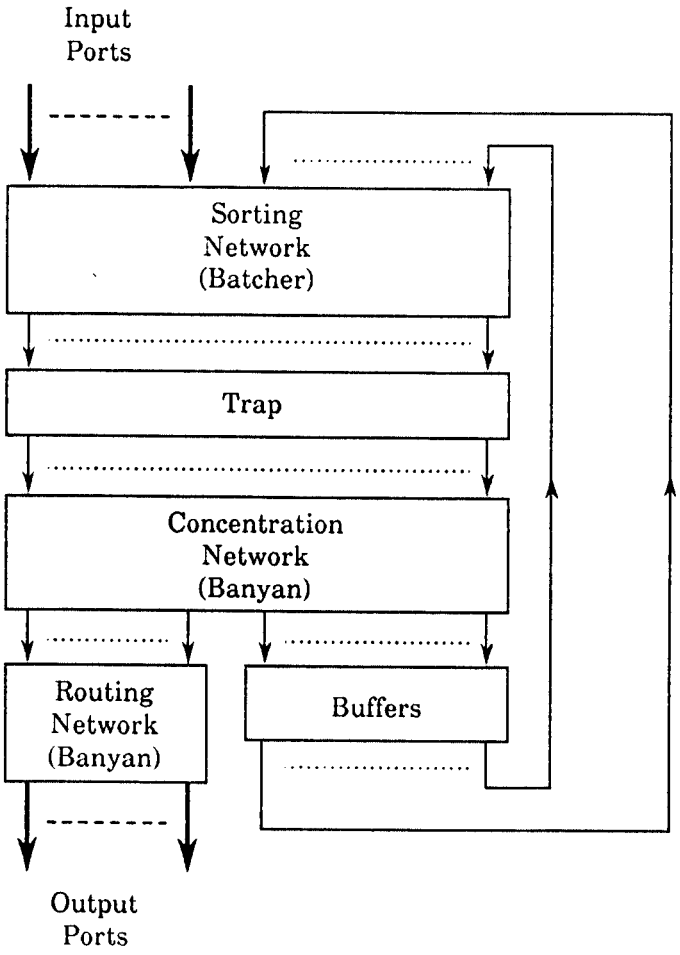

Figure 3.4: The Starlite fast packet switch.

3.3 Output Buffered Switches

Starlite

This is another fast packet switch design based upon the use of a Batcher-banyan switch fabric but overcomes the head of the line blocking problem by storing packets which fail arbitration in shared buffers at the output of the sorting network rather than at the input [70]. The basic structure of the Starlite switch is shown in fig. 3.4. Again the switch is synchronous at the packet level. On entry to the switch a tag is prefixed to each packet which contains the port number of the required output port. Incoming packets are sorted according to the tag by the Batcher network and the trap arbitrates between packets contending for the same output port which appear on adjacent ports on output from the Batcher network. No more than one packet may be directed to any single output port during the same timeslot thus conflicting packets which lose arbitration must be directed to the buffers in order to

be recirculated through the network during the next timeslot. To accomplish this the trap computes a running sum address for those packets destined for the output ports and also for those to be directed to the recirculation buffers and appends this address as a further tag onto the front of each packet. When the packets then pass through the concentration network, packets destined to the output ports emerge on the left hand outputs, packets directed to the recirculation buffers on the right hand outputs, and empty packets in the centre. Packets directed to the output ports are routed to the requested ports by the banyan routing network while packets which lost arbitration are stored in a pool of shared buffers and re-enter the sorting network together with incoming packets at the next timeslot.

The same comment as regards difficulties with the size and partitioning of the Batcher sorting network applies to the Starlite design as to the three phase Batcher-banyan but here we also have the running adders of the trap network and an additional banyan concentration network to include. Thus more stages are required in the switch fabric and the number of interconnections is increased. In addition to this, analysis shows that the number of switch fabric ports devoted to re-entry must be two to three times the number of the switch input/output ports to ensure an acceptably low probability of packet loss, [65, 34]. The additional hardware required in the sorting network to handle the recirculated packets is reduced by the fact that the recirculated packets have already been sorted into order. However, a reduction in the buffering requirements is gained at the expense of a much larger switch fabric when compared to other switch designs. One further difficulty with the Starlite switch is that packets may be delivered out of sequence due to the recirculation mechanism. This may be overcome with a simple time stamping technique which gives older packets a higher priority at arbitration.

An extension to the Starlite switch to offer multicast capability is discussed in [70] which requires another two interconnection networks to be added to the switch fabric. Empty packets are injected into the switch fabric directed to every output in the multicast groups and are filled with a copy of the required packet data field in a copy fabric. Another modification of the basic Starlite design is discussed in [34] which uses multiple adjacent re-entry loops. An example of the layout of a 128×128 switch reveals a very large number of switch stages and interconnections.

Finally, it is a simple matter to extend any switch based upon the Batcher-banyan switch fabric to handle any number of levels of packet priority. If the priority field is appended to the least significant digit of the destination field in the tag at the front of the packet, then all packets contending for the same output port will emerge from the Batcher network sorted into order of priority. The arbitration logic may thus easily select the highest priority packet. This is likely to become a desirable feature of a fast packet switch in any network handling multi-service traffic but particularly in a public network.

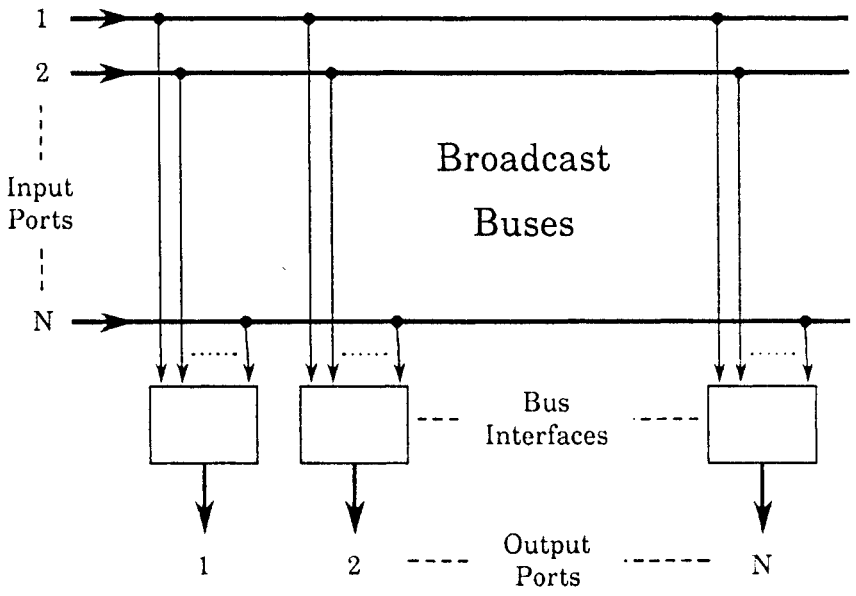

Figure 3.5: Structure of the Knockout Switch.

The Knockout Switch

The Knockout Switch is another example of an output buffered switch but differs from Starlite in that buffers are dedicated to specific output ports and not shared [163]. There is therefore no recirculation of packets. Packets which cannot be accommodated by the available buffers are discarded and the switch is dimensioned to keep the probability of packet loss sufficiently low.

The structure of the Knockout Switch is outlined in fig. 3.5. The switch is based on a fully connected switch fabric in which the traffic on each input port is broadcast to every output port. The switch operates synchronously at the packet level and at the start of each timeslot, incoming packets are broadcast across the buses of the switch fabric preceded by their destination tag. The bus interfaces of every output port inspect the destination tags of every packet on all of the broadcast buses to select packets that are addressed to them. These packets are then buffered in the switch output ports provided that no more than a limited number, typically eight, arrive at any output port during the same timeslot. The construction of the bus interface is shown in fig. 3.6. The packet filters select those packets addressed to the output port to which they belong. These are then submitted to an N input L output concentration network. Provided that no more than L packets arrive in any given timeslot they are passed on to the shared buffer otherwise excess packets are discarded. The shared buffer allows up to L packets to arrive simultaneously but maintains a single first in first out queue by means of the shifter which allocates packets to buffers in a cyclic manner. The buffers are emptied in a cyclic manner.

The most obvious difficulty with the Knockout Switch arises from the use of a fully

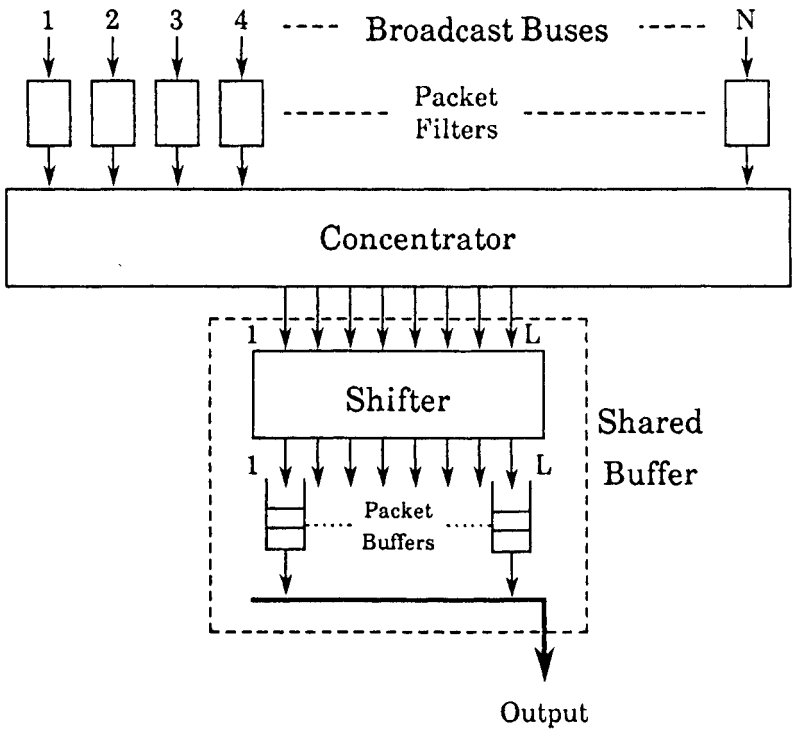

Figure 3.6: The bus interface of the Knockout Switch.

interconnected broadcast switch fabric. This results in an increase in the hardware and interconnections required of at least one order of magnitude when compared to input buffered switches of the same size. It does, however, offer the advantage of incremental growth allowing the switch to grow by one port at a time. The buffer requirement is approximately twice that of an input buffered switch for the same packet loss probability. Eight parallel buffers, each of five packets deep, in each switch output port will yield a packet loss probability of 10^{-6} at a load of 84% for random traffic. This is considered by the designers to be acceptable when compared to packet loss from other sources.

An extension of the Knockout Switch to handle variable length packets is introduced in [44] together with further thoughts on the possible implementation of the switch. Implementation of the Knockout switch using photonic components in the data paths of the switch is discussed in [43]. The broadcast nature of the switch fabric is likely to limit the maximum operating speed of an electrical implementation of the switch and will limit the maximum size of a photonic implementation. The $O(N^2)$ interconnection requirement will make switches larger than a few hundred ports difficult to construct. The broadcast nature of the switch fabric might suggest that the Knockout Switch could easily be adapted for multicast operation but this would require excessive complexity in the packet filters. Some suggestions have been made in [45] regarding the support of multicast capability but only at a fairly low

multicast capacity. The performance of an output buffered switch for periodic input traffic is considered in [75].

3.4 Internally Buffered Switches

An internally buffered fast packet switch is constructed from switching elements that contain one or more packet buffers per port. As with the construction of non-buffered switches we find that buffered switching elements may be divided into two classes: output buffered switching elements and input buffered switching elements. As the name suggests, an output buffered switching element is constructed by placing the buffering after the switching function on the outputs of the switching element. This gives the output buffered switching element a higher throughput than for an input buffered switching element due to head of the line blocking within the input queues of the input buffered switching element. It is, however, a more complex device to implement and requires a large number of packet buffers on each output port to reduce packet loss to reasonable proportions. Some proposed designs of output buffered switching element tend towards being complete fast packet switches in their own right, with a capacity of several Gbits/sec, of degree 8 or 16. In general, three-stage networks are suggested for the switch fabric structure but little work is available on the performance of such buffered switching elements in large switch structures.

In contrast, the input buffered switching elements suggested in the literature are small (2×2), with limited buffering of one or two packet buffers on each of the input ports of the switching element. They rely on backpressure between the stages of the switch fabric to prevent buffer overflow. They are based upon the banyan switch fabric, possibly with additional stages to distribute incident traffic evenly across the switch fabric, for which the performance is a little better characterised than for the class of output buffered switching elements.

Output Buffered Switching Elements

Prelude

Prelude, together with Starlite, is amongst the earlier designs of fast packet switch [141, 31]. As such it more closely resembles the structure of the conventional shared memory digital circuit switch than other fast packet switch designs. It forms an output buffered 16×16 switching element in which all of the functions of a complete fast packet switch are implemented. The switching element operates at a bit rate of 280 Mbits/sec on each switch port, with packets of length 128 bits, giving a capacity of about 3.6 Gbits/sec for a switching element operating at a load of 80%.

A serial to parallel converter transforms the incoming serial bit stream on each of the input ports into a stream of 8 bit octets in parallel, fig. 3.7. With 16 input ports and a total packet length of 16 octets, including a one octet header, a delay is introduced so as to align each of the incoming streams such that only a single packet

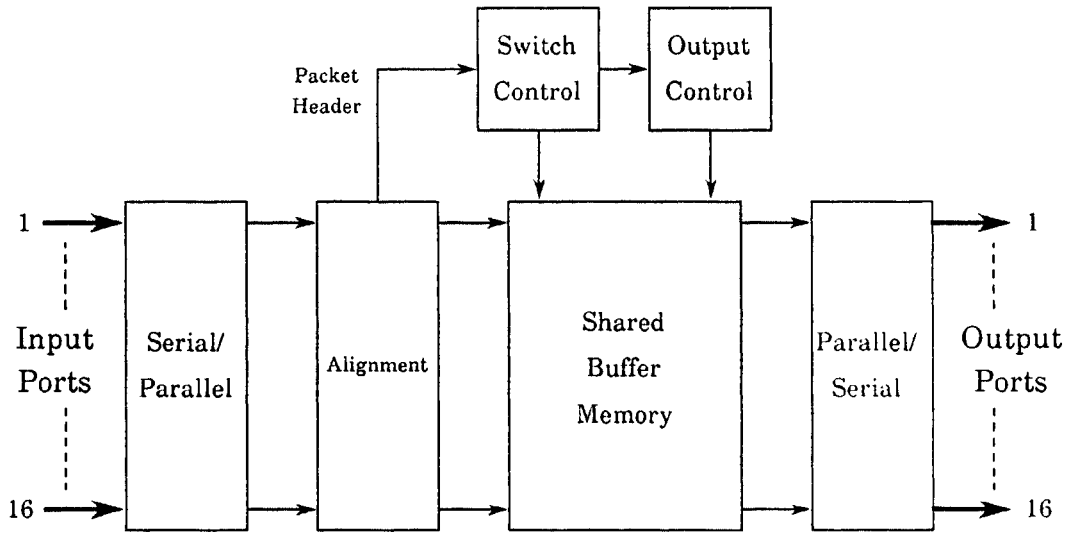

Figure 3.7: Structure of the Prelude switching element.

header emerges from the alignment unit during each octet clock cycle. A translation operation by table look-up is performed on the 8 bit packet header and the packet allocated a position in the shared memory in which it is to be stored. At the same time an entry is made in the relevant FIFO output queue of the output port to which it is routed. These output queues, one for each output port, consist of pointers to the location of the relevant packets in the shared memory. On the output side of the switch, packets are extracted from the shared memory and transmitted over the output ports according to the entries in the output queues.

The use of a shared memory, with a single shared processing unit for the packet headers, may limit the flexibility of the Prelude design. It does, however, allow a large amount of buffer space to be shared dynamically between all output ports. The switch could be extended to offer two levels of packet priority without great difficulty. Also the switch may offer multicast connections by writing pointers to the same packet into multiple output queues, although this introduces the possible problem of knowing when a packet may be deleted from the shared buffer space.

The Bus Matrix Switching Element

In common with the Prelude design, the Bus Matrix Switch implements all of the functions of a complete fast packet switch in every switching element [120]. The structure of the switching element is given in fig. 3.8. The primary packet distributor (PPD) inspects the incoming packets and transmits them across a bus to the appropriate cross point memory (XPM). The cross point memory is a FIFO queue which connects the incoming buses from the PPDs to the outgoing buses and these are arranged in a matrix structure. The secondary packet distributors (SPDs) read out packets from the cross point memories to the output ports.

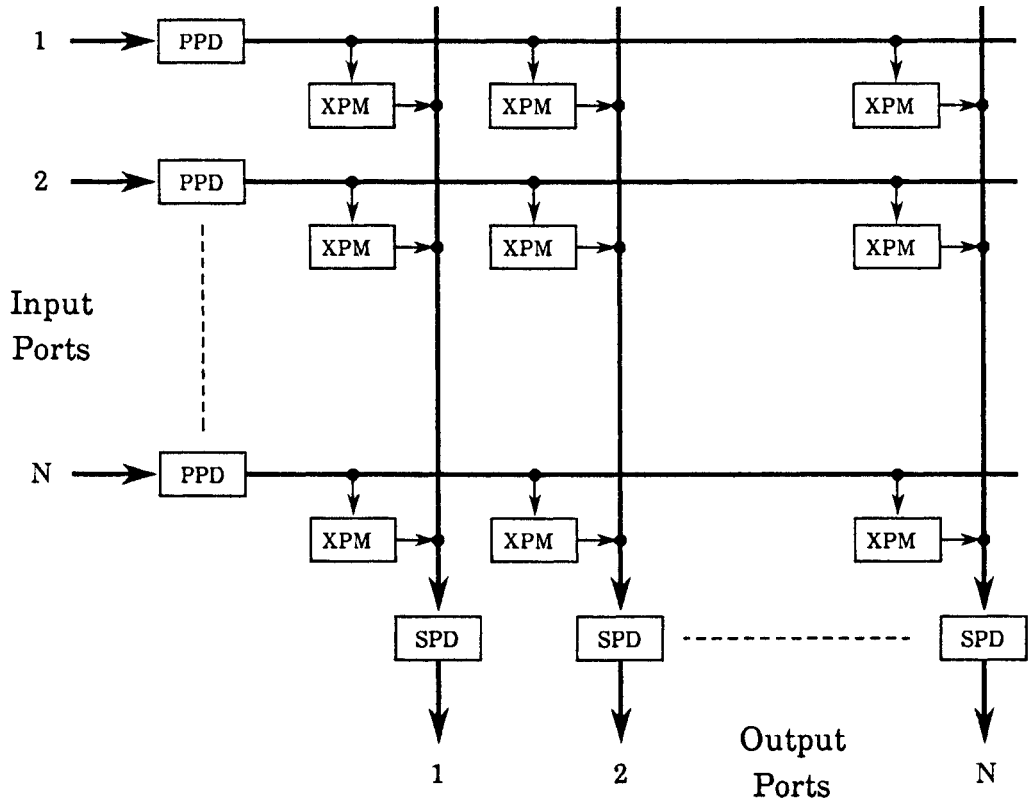

Figure 3.8: Structure of the Bus Matrix switching element.

The switch operates asynchronously at the packet level and may handle variable length packets. It is claimed that the broadcast function is easily implemented though multicast operation would probably be much more difficult. A size of 16 Kbytes is proposed for each cross point memory which suggests that using VLSI a 16×16 switching element could be implemented on a single circuit board. For CMOS implementation a maximum switching element size of 16×16 may be achieved with each input port operating at 160 Mbits/sec using 8 bit wide internal paths for the bus matrix. This would offer a total capacity of 2 Gbits/sec at a loading of 80%. For an ECL implementation the maximum switching element size is 8×8 with a line bit rate of 800 Mbits/sec offering 5 Gbits/sec switch capacity at an 80% load. A three stage switch fabric structure is suggested to achieve larger sizes of switch but little detailed information is given.

This design of buffered switching element is flexible in that various sizes and capacities of switching element may be constructed from standard parts. It does, however, require much more buffering than other designs as buffer space is partitioned not only between the outputs but also according to each input, thus there is no sharing of buffer space whatsoever.

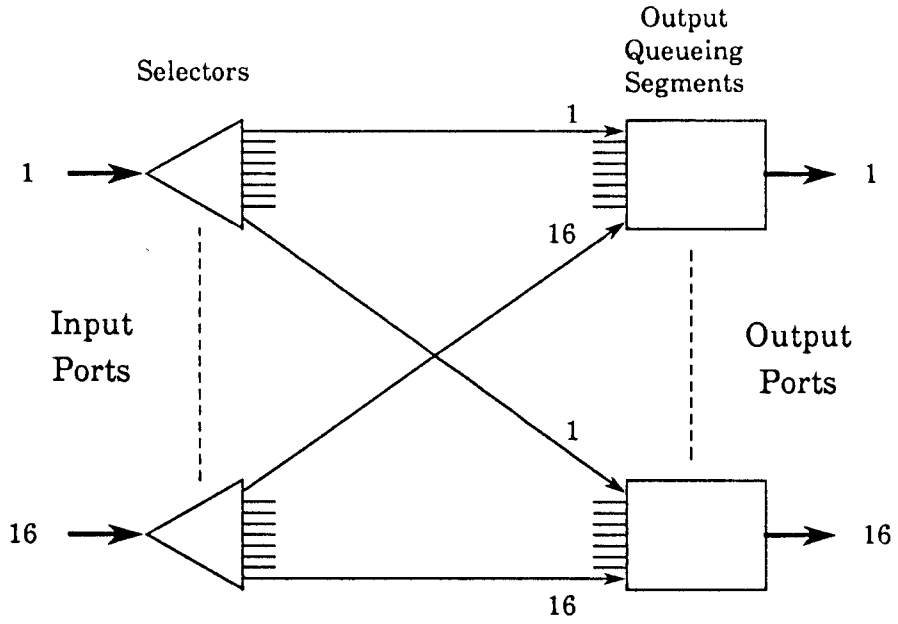

Figure 3.9: Structure of the IBM switching element.

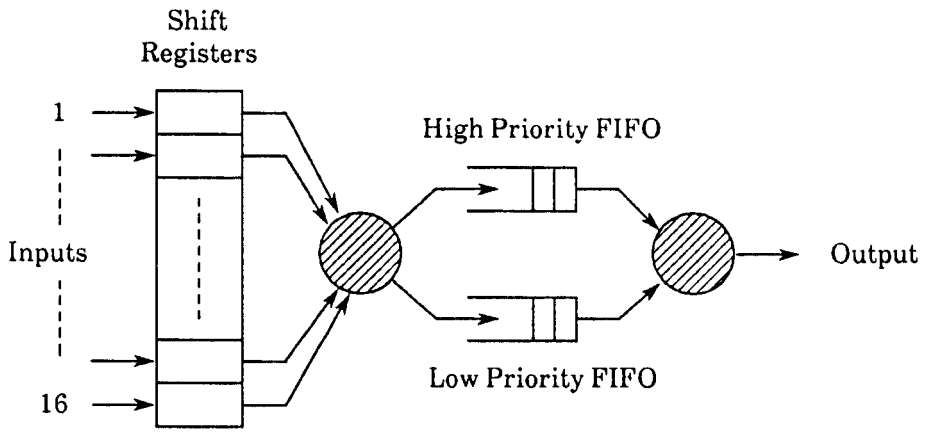

Figure 3.10: The output queueing segment of the IBM switching element.

The IBM Switching Element

This is a 16×16 output buffered switching element designed to offer a bit rate of 32 Mbits/sec on each switch port with two levels of packet priority to accommodate traffic with real-time constraints [4]. The structure of the switching element is illustrated in fig. 3.9. The selector examines the tag at the head of an incoming packet and forwards it to the corresponding output queueing segment. The output queueing segment, shown in fig. 3.10, is capable of accepting 16 packets arriving at once and queueing them in either of the two output FIFO queues according to the

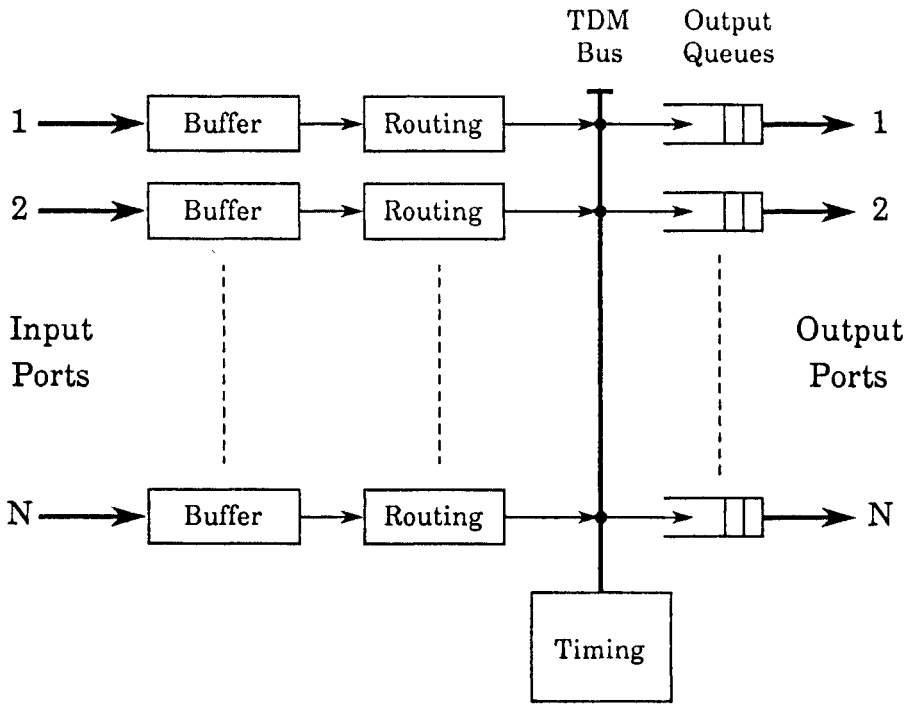

Figure 3.11: Structure of the TDM Bus switching element.

packet priority. It achieves this by operating at 16 times the line rate of the switch ports and uses a set of 16 shift registers to add a small amount of temporary storage. The design of the output queueing segment to operate at 16 times the line rate of the switch ports limits the bandwidth of the switch ports that may be attained. Also, large output FIFO queue sizes, in excess of a hundred packets per port, are proposed to maintain a low packet loss rate which might prove expensive to implement in VLSI. A backpressure mechanism is discussed to reduce the length of the output queues. Suggestions are given for the use of the switching element in both single stage and three stage switch fabrics.

The TDM Bus Switching Element

The last example of a high capacity switching element, [36, 35], bears some resemblance to the IBM switching element. An outline of the structure of the switching element is provided in fig. 3.11. Incoming packets are buffered in the single packet buffer at each of the input ports of the switching element. The TDM bus operates at the sum of the line rates of the input ports and each input port has a corresponding timeslot. The routing logic directs the packet to the required output queue, according to the tag at the head of the packet, during its timeslot on the bus. The output queues must also work at the same rate as the TDM bus in order to provide non-blocking operation.

The size of the switching element will be limited by the speed of the TDM bus and output queues, 8×8 at 560 Mbits/sec being suggested as possible within a few years. The output queues will also have to be large to minimise the packet loss due to buffer overflow, a length of 40 packets being proposed. The packet length is fixed, switch operation is synchronous and multicast operation within the switching element may be difficult to implement. As with the IBM switching element, packet priority could be introduced at the cost of increasing the number of output queues.

Input Buffered Switching Elements

Several fast packet switch designs have been proposed based upon the use of a 2×2 input buffered switching element of which two offer some details of the implementation of the switching element, Turner [145, 148, 19] and CSELT [128, 56]. In both cases the switching element is implemented in CMOS running at about 25 MHz with 8 bit wide data paths. Turner suggests the use of large buffers of about 10 Kbits each while CSELT, who have implemented their switching element in 3 μm CMOS, have used 512 bits per input buffer. Both designs employ backpressure between adjacent switch stages to avoid buffer overflow. A banyan interconnection network is suggested for the switch fabric with additional stages added to distribute the traffic evenly across the switch fabric. The CSELT switching element has been implemented using 6,000 gates which demonstrates the hardware simplicity of an input buffered switching element when compared to output buffered switching elements. Turner goes to some detail in considering the support of multicast connections across his switch design.

Although an input buffered switching element provides a much simpler design it also offers a lower performance than that of the output buffered switching element. Also the 2×2 elements require many more switching stages within the switch fabric and thus more interconnections than a switch fabric constructed from switching elements of higher degree. This tends to reduce the maximum size and capacity of switch which may be constructed.

3.5 Performance Comparison

Some general analytical and simulation results are available which allow a first order comparison of the performance of the basic switch structures. An analytical investigation of input buffered interconnection networks was presented in [126] which looks at the throughput at saturation of the non-blocking (or crossbar) network and also at the banyan (or delta) network. The result for the banyan network is expressed as a recurrence relation for which an asymptotic analysis is presented in [82] and upper and lower bounds on the solution in [85]. These analytical solutions, however, simplify the problem by assuming that blocked packets are discarded and that the switch is operated synchronously at the packet level. The first assumption causes the throughput to be overestimated while for the banyan network the second assumption leads to an underestimation because blocking within the network is maximised by

the assumption of synchronisation. Analysis of the throughput at saturation of input buffered, non-blocking switch fabrics in which blocked packets are not discarded but are queued and resubmitted is offered in [76] and [71]. A comparison of the delay performance of both input and output buffered switch fabrics is offered in [76]. The outcome of the above analyses is to demonstrate that for the non-buffered switch fabric an input buffered, non-blocking switch has a throughput at saturation of approximately 58% that of an output buffered switch. This is due to the effect of head of the line blocking within the input queue of a pure input buffered switch.

Some results are also available for an internally buffered switch fabric constructed from 2×2 switching elements. An analytical approach is taken in [40, 41, 74] while [100, 19] offer simulation results. Differences in the models of the switching elements mean that the results are not in exact agreement but in general they show that buffered switch fabrics of 2×2 switching elements with a single buffer on each input port offer a throughput at saturation performance similar to input buffered switch fabrics of non-buffered 2×2 switching elements. Further, internally buffered switch fabrics constructed from 2×2 switching elements with four buffers at each input port offer a throughput at saturation performance comparable to non-blocking, input buffered switch fabrics.

3.6 Summary

A fast packet switch requires two fundamental components: switching and buffering. Switching takes place in the switch fabric which is generally a multi-stage interconnection network constructed from the interconnection of a large number of fundamental switching elements in stages. If the buffering is external to the switch fabric a non-buffered switch fabric results, formed from non-buffered switching elements. In such a switch design, if the buffering precedes the switch fabric an input buffered switch results. Else, if the buffering follows the switch fabric an output buffered switch results. An internally buffered switch fabric is composed of buffered switching elements. The buffering within a switching element may either be located at its input ports or at its output ports.

An input buffered switch with a non-blocking switch fabric offers approximately 58% of the throughput at saturation performance of an output buffered switch. A blocking switch fabric will offer even lower performance but techniques such as input queue by-pass and multiple switch planes with output buffering across the switch planes exist and will increase the performance. An output buffered switch requires much more hardware than an input buffered switch of comparable size.

The performance of an internally buffered switch fabric composed of switching elements of high degree with buffering on the output ports will offer a performance which approaches that of an output buffered switch fabric. The switching element will, however, be complex, require much hardware for implementation and will require large buffer sizes. An internally buffered switch fabric composed of switching elements of low degree with small buffer sizes on the input ports and backpressure between

switch stages will exhibit a performance comparable to that of the input buffered switch fabric.