A Survey of Modern High-Performance Switching Techniques

HAMID AHMADI, MEMBER, IEEE, AND WOLFGANG E. DENZEL

Abstract—**The rapid evolution in the field of telecommunications has led to the emergence of new switching technologies to support a variety of communication services with a wide range of transmission rates in a common, unified integrated services network. At the same time, the progress in the field of VSLI technology has brought up new design principles of high-performance, high-capacity switching fabrics to be used in the integrated networks of the future. Most of the recent proposals for such high-performance switching fabrics have been based on a principle known as fast packet switching. This principle employs a high degree of parallelism, distributed control, and routing performed at the hardware level. In this paper, we present a survey of high-performance switch fabric architectures which incorporate fast packet switching as their underlying switching technique to handle various traffic types. Our intention is to give a descriptive overview of the major activities in this rapidly evolving field of telecommunications.**

INTRODUCTION

THE existing telecommunications networks, whether circuit-switched or packet-switched, are oriented towards particular applications. Thus, we have different networks for voice, signaling, video, and data applications operating in parallel and independently. While each of these networks is suitable for the application it is designed for, they are not very efficient for supporting other applications. The advantages of an integrated communication system which can accommodate a variety of diverse services with different bandwidth requirements has been recognized for some time [1]–[5]. Reasons of economy and flexibility motivated its development. The objective of having a unified integrated network is the flexibility to cover existing as well as future services with good performance and economical resource utilization, and with a unified network management, operation, and maintenance [6], [7].

The rapid evolution in the field of telecommunications, especially in the area of transmission systems and fiber optics, has led to new switching technologies for integrated networks. Advances in the field of VLSI technology while reducing the cost of circuit fabrication have brought about completely new principles in the design and architecture of high-performance switching fabrics which can accommodate a wide range of bandwidths. The intention of this paper is to give a survey of the state of the art and characteristics of these switching technologies.

The traditional circuit switching concept evolved as a way to handle and transport stream-type traffic such as voice and video. A circuit-switched connection is set up with a fixed bandwidth for the duration of a connection which provides fixed throughput and constant delay. The classical switching techniques employed for this type of switching are *space division multiplexing* (SDM), *time division multiplexing* (TDM), and combinations of both techniques. In pure SDM switching arrangements, there is no multiplexing of connections on the internal data paths or space segments of the switch. In systems which employ the TDM technique, multiple connections are time multiplexed on the internal data paths of the switch. A controller schedules the allocation of time and/or space segments in advance in order to avoid any conflict. While this technology has been very attractive and familiar, it is less efficient in supporting different bit rates that are needed for various services. Even in the multirate circuit-switching systems which allow the allocation of bandwidth equal to integer multiples of some basic rate, as Kulzer and Montgomery point out [8], the choice of a basic rate is a difficult engineering design decision. In order to accommodate the relatively low basic ISDN rate, many parallel low-rate channels must be established for high-rate services. This will imply extra control overhead needed to set up these parallel channels. In addition, all channels that comprise a single connection must be synchronized with no differential delay within each channel.

Packet switching evolved primarily as an efficient way to carry data communications traffic. Its characteristics are access with buffering, statistical multiplexing, and variable throughput and delay, which is a consequence of the dynamic sharing of communication resources to improve utilization. Today, it is used exclusively for data applications, and it employs both virtual circuits and datagram techniques. Very sophisticated protocols have been developed which incorporate complex functions, such as error recovery, flow control, network routing, and session control. The switching capacity of current conventional packet switches ranges from 1 to 4 thousand packets per second [9], [10] with average nodal delays of 20–50 ms.

Manuscript received July 13, 1988.

H. Ahmadi is with the IBM Research Division, T. J. Watson Research Center, Yorktown Heights, NY 10598.

W. E. Denzel is with the IBM Research Division, Zurich Research Laboratory, 8803 Rüschlikon, Switzerland.

IEEE Log Number 8929565.

The switching functions are typically performed by means of software processing on a general-purpose computer or a set of special-purpose processors. The capabilities of these packet switches are very attractive for applications requiring low throughput and low delay such as inquiry/response-type traffic and those requiring high throughput, but which can tolerate higher delays such as file transfer. However, these capabilities are not sufficient for real-time types of traffic such as voice, video, or computer-to-computer data transfer.

The rapid pace of technological change has brought about new switching system concepts in order to satisfy the high-performance requirement for future systems. Many different switch fabric designs have been proposed and developed at various research organizations around the world over the past few years. All current approaches of high-performance switching fabrics employ a high degree of parallelism, distributed control, and the routing function is performed at the hardware level. In this paper, we will survey many of these architectures. Our intention is to give a descriptive overview of the major recent activities in this rapidly evolving field of telecommunications.

Some recent switching approaches have been based on the *Fast Circuit Switching* (FCS) concept. This idea relies on the fast setting up and taking down of connections such that the system does not allocate any circuit to a user during his idle time. An architecture for integrated data/voice with fast circuit switching was proposed by Ross *et al.* [11]. A form of a distributed fast circuit switching which is called "burst switching" has been reported in [12]–[14].

Our emphasis in this paper is on those switch fabrics which incorporate the *Fast Packet Switching* (FPS) concept [6], [7], [15], [16] as their underlying switching technique to support a wide range of services with different bit rates. We try to categorize and classify them according to their internal fabric structures. In this regard, we will classify them into the following categories:

- Banyan and buffered banyan-based fabrics
- Sort-banyan-based fabrics
- Fabrics with disjoint-path topology and output queueing
- Crossbar-based fabrics
- Time division fabrics with common packet memory
- Fabrics with shared medium.

In the following sections, we give a detailed descriptive overview of most switch architectures within each class.

Banyan and Buffered Banyan-Based Fabrics

The very early theoretical work on *multistage interconnection networks* (MIN) was done in the context of circuit-switched telephone networks [17], [18]. The aim was to design a nonblocking multistage switch with the number of crosspoints less than a single-stage crossbar matrix. Later on, many multistage interconnection networks were realized and studied for the purpose of interconnecting multiple processors and memories in parallel computer systems. Subsequently, several types of these networks such as *banyan* and *delta* networks have been proposed [6]–[8] as a switching fabric for integrated telecommunication switching nodes.

Banyan networks, which were originally defined by Goke and Lipovski [19], belong to a class of multistage interconnection networks with the property that there is exactly one path from any input to any output. Banyan networks are subdivided into several classes [20]. Of practical interest are the regular, rectangular SW-banyans which are constructed from identical switching elements with the same number of inputs as outputs.

Delta networks as defined by Patel [21], [22] are a subclass of banyan networks which have the digit-controlled routing or self-routing property. A rectangular $N \times N$ delta network, called *delta-b*, is constructed from identical $b \times b$ switching elements or nodes in k stages where $N = b^k$.

Many of the well-known interconnection networks such as omega, flip, cube, shuffle-exchange, and baseline belong to the class of delta networks [22]. These networks have been considered for packet-switching techniques to obtain high throughput because several packets can be switched simultaneously and in parallel and the switching function is implemented in hardware. Although these networks have different interconnection patterns, they have the same performance in a packet-switching environment.

The principal characteristics of these networks are: 1) they consist of $\log_b N$ stages and N/b nodes per stage, 2) they have the self-routing property for packet movements from any input to any output by using a unique k digit, base b destination address, 3) they can be constructed in a modular way from smaller subswitches, 4) they can be operated in a synchronous or asynchronous mode, and 5) their regularity and interconnection pattern are very attractive for VLSI implementation. Fig. 1 shows an example of an 8×8 delta-2 network with the bold lines indicating the routing paths of two packets.

While these networks are capable of switching packets simultaneously and in parallel, they are blocking networks in the sense that packets can collide with each other and get lost. There are two forms of blocking: internal link blocking and output port blocking. The internal link blocking refers to a case where packets are lost due to contention for a particular link inside the network. The output port blocking, however, occurs when two or more packets are contending for the same output port. Fig. 2 shows these effects in an example of an 8×8 delta-2 network. These two effects result in the reduction of the maximum throughput of the switch. The performance of these switches has been studied extensively [22]–[27]. Reference [25] gives a survey on the performance of these networks in a packet-switching environment. This reference also gives detailed attention to how the performance of these networks can be improved.

There are several ways to reduce the blocking or to increase the throughput of banyan-type switches:

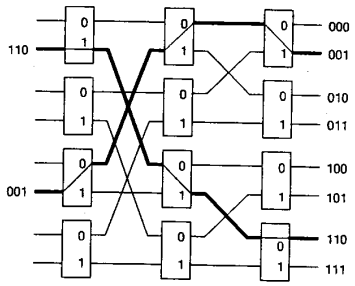


Fig. 1. An 8 × 8 delta-2 network.

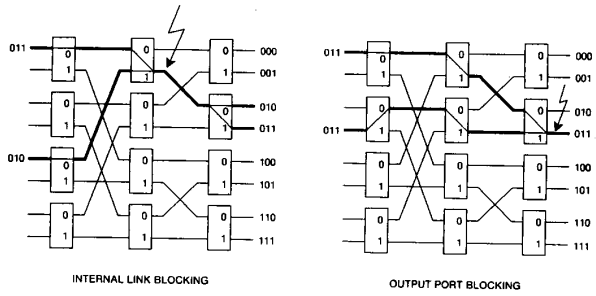


Fig. 2. Internal link blocking and output port blocking in an 8 × 8 delta-2 network.

- increasing the internal link speeds relative to the external speeds,
- placing buffers in every switching node,
- using a handshaking mechanism between stages or a back-pressure mechanism to delay the transfer of blocked packets,
- using multiple networks in parallel to provide multiple paths from any input to any output or multiple links for each switch connection [26], [27],
- using a distribution network in front of the banyan network to distribute the load evenly.

The idea of using a *buffered* delta network for switching in multiprocessor systems was first introduced and analyzed by Dias [23]. This idea was further developed and considered for a general-purpose packet switching system by Turner [6], [28], [29]. He proposed a buffered banyan switch architecture which led to the dawn of research activities on fast packet switching [6], [28]. He further developed and advanced the concept to the *Integrated Services Packet Network* (ISPN) architecture [29]. The ISPN architecture is based on a large high-performance packet switch structure as illustrated in Fig. 3. The switch interfaces up to 1000 high-speed digital transmission facilities (e.g. 1.5 Mbits/s) via packet processors (PP). The packet processor provides input buffering, adds the routing header, and performs the link level protocol functions. A control processor (CP) performs all connection control functions. The switch fabric is based on a ten-stage self-routing buffered banyan network with 1024 ports made up of 5120 binary switching elements. Each 2 × 2 switching element has a buffer at each input port capable of storing one packet. The internal links joining the nodes are bit

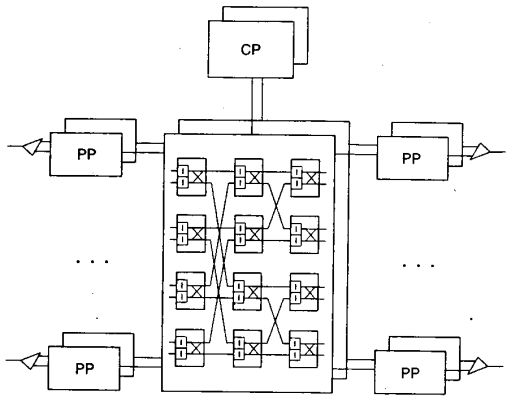


Fig. 3. Turner's ISPN packet switch structure [7].

serial and operate at eight times the speed of the external links. This means that a 100 percent load on the external links yields a 12.5 percent load on the internal switch fabric links. The switch fabric also uses a back-pressure flow control mechanism between stages which prevents buffer overflow or packet loss within the fabric. The buffering technique proposed is based on the *virtual cut-through* concept described by Kermani and Kleinrock [30]. When a packet arrives at a switching element and the desired port is free, it is directly sent through, bypassing the buffer. This way, the delay through the switching elements is reduced to a minimum.

A *Wideband Packet Technology* network with packet switches based on Turner's original buffered banyan development and capable of switching packetized voice, data, and video has been reported by AT&T [31], [32]. A field trial was carried out between several AT&T locations in California in order to demonstrate the feasibility of a fully integrated packet network working in conjunction with existing transmission facilities. The experimental 16 × 16 buffered banyan fabric has a clock rate of 8 Mbits/s and the external transmission line speed is 1.5 Mbits/s. Each binary switching element has a buffer for one complete packet per input. Recently, the same concept has also been proposed for use at much higher speeds, realized in GaAs technology [33].

A high-performance *Broadcast Packet Switching Network* has been proposed recently by Turner [34]. The principle is based on the ISPN architecture described above, but with higher bandwidth and broadcasting capability. The switch architecture consists of a 64 × 64 buffered banyan fabric as a routing network (RN), preceded by a copy network (CN) and a distribution network (DN); see Fig. 4. The distribution network is itself a buffered banyan network and its purpose is to reduce internal link blocking by distributing the traffic offered to the routing network evenly across all input ports. The nodes of the distribution network ignore the destination addresses of the packets and route them alternatively to both node outputs. The copy network also has a banyan structure. Each node, when it receives a broadcast packet (which

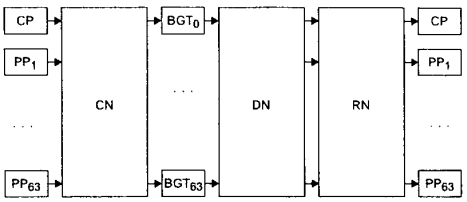


Fig. 4. Basic structure of Turner's Broadcast Packet Switching Network. From [7], © 1986 IEEE.

has a header field indicating the number of copies), copies the packet, and sends it out on both outgoing links and modifies the "number of copies" field. Between the outputs of the copy network and inputs of the distribution network are the broadcast and group translators (BGT). The function of the BGT's is to translate the destination address of the packet copies which emanate from the copy network in order to be routed to different destinations. Thus, they translate the group number and logical channel number of the broadcast packets into an output port number. For details of the copy network routing algorithm, the reader is referred to [29].

The performance of buffered banyan networks has been studied extensively [23]–[27], [35]–[38]. The analytical model presented by Jenq [35] is for a packet switch based on a delta-2 network with single buffers in each node. The result of this model shows that, for a balanced and uniform traffic pattern, the maximum throughput of the switch is about 40 percent. Extensive simulation studies of the buffered banyan network, specifically in the context of the ISPN switch architecture described above, is presented in [36]. This study is different from others in several respects. It considers the performance of the routing network with emphasis on the effect of cut-through switching, different buffer sizes within each node, and the various node sizes. In addition, it also investigates the effect of nonuniform traffic on the routing network and the improvements achieved by the distribution network in front of the routing network. The performance of the copy network for broadcasting packets is also given in this report. In a recent paper [37], an approximate analytic approach based on a Markov chain model is presented for a general delta-b network with multiple buffers within each node. Numerical results in this report as well as the results in [36] indicate that, by increasing the size of the nodes and the buffer size, respectively, the maximum throughput of the switch can be increased to about 60 percent. However, this improvement is not linear. In fact, for a buffer size of more than four packets per port or a node size greater than 4×4, the amount of improvement in throughput is not very significant.

In these studies of buffered banyan networks, the position of the buffers within each switching element has been at the inputs. Recently, buffered banyan structures have also been considered which use switching elements of larger size and have buffers at the module outputs [38], [54]. We will defer the discussion about this architecture until Section IV.

Sort-Banyan-Based Fabrics

As mentioned before, one drawback of the banyan networks is that they are internally blocking in the sense that two packets destined for two different outputs may collide in one of the intermediate nodes. However, if packets are first sorted based on their destination addresses and then routed through the banyan network, the internal blocking problem can be avoided completely. This is the basic idea behind the *sort-banyan* type networks. Fig. 5 shows the basic structure of a sort-banyan network. The first segment consists of a sorting network (in this case, a Batcher bitonic sort network [39]) which sorts the packets according to their destination address, followed by a shuffle exchange and a banyan network which routes the packets. It can be shown that packets will not block within the banyan network. Note that Fig. 5 shows two connection patterns which would have caused internal link blocking in a pure banyan network (see Fig. 2). Each node of the sort network as shown in Fig. 5 is a 2×2 switching element which sorts the incoming packets in the order as indicated by the arrow shown. The number of elements in the sorting segment is $N/4((\log_2 N)^2 + \log_2 N)$. The network operates in a synchronous manner, and packets are processed in each stage in parallel. While this self-routing interconnection network is internally nonblocking, blocking still may occur if destination addresses are not distinct, i.e., if there are packets with identical addresses. Hence, simultaneous packets destined for the same output will collide within the banyan part of the network.

The very first switch fabric implementation which proposed and employed the sort-banyan self-routing structure is the *Starlite* switch [40]; see Fig. 6. At the switch interface, various services are converted into constant length "switch packets" with a packet header indicating the routing information which is the destination address. To overcome the output port contention problem, the Starlite approach uses a trap network between the sort and the banyan (called the expander here) segment which detects packets with equal destination addresses at the output of the sort segment. Thus, the packets with repeated addresses are separated from the ones with distinct addresses. The packets with repeated addresses are fed back to the input side of the sort network for reentry within the next cycle. These packets can only use the idle input ports. Since the number of recycled packets at any time can be larger than the free input ports, a buffering stage must be used for the recycled packets if packet loss is not acceptable. The trap network consists of a single-stage comparator followed by a banyan network of the same type as the routing network. Note that if packets were simply recycled through the switch, they would be delivered out of sequence. This problem is solved by "aging" packets as they recirculate and by using the sort network to give old packets priority over new ones. The Starlite switch has also been considered for optical implementation [41].

Recently, Hui [42], [43] proposed a switch architecture

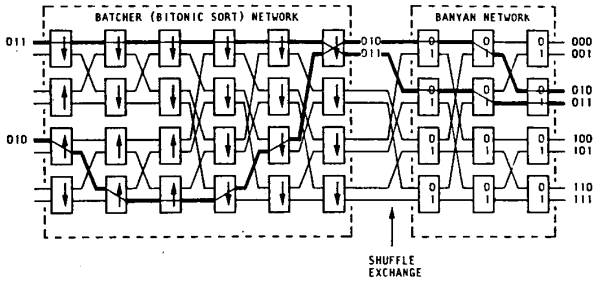


Fig. 5. Basic structure of a sort-banyan network.

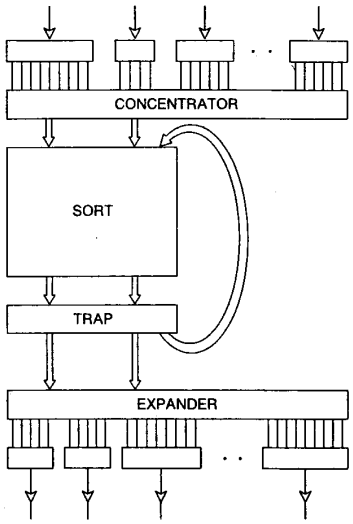


Fig. 6. Basic structure of the Starlite switch. From [40], © 1984 IEEE.

based on the sort-banyan network structure, but with a different scheme than Starlite to overcome the output port contention problem. The advantage of this approach is that the switch fabric delivers exactly one packet to each output port from one of the input ports which request a packet delivery to the same output port. Hence, packets are never lost within the fabric and they are delivered in sequence.

The basic structure of this fabric, like the Starlite switch, is a Batcher sorting network followed by a banyan routing network. As mentioned before, the combined sort-banyan network is internally nonblocking. To resolve the output port conflict, a three-phase algorithm in conjunction with input queueing at each input port is employed. All packets are of a constant length and the switch operates in a synchronous manner. In the first phase [Fig. 7(a)], which is the arbitration phase, each input port which has a packet in its queue for transmission sends a short request packet. The request packet carries just the source/destination address pair. The requests are sorted in ascending order by the Batcher sorting network according to the destination address. All but one request for the same destination are purged. The second phase [Fig. 7(b)] is to acknowledge the input ports which have won the arbitration since the input ports which made the requests do not know the result of the arbitration. In this phase, the winning requests are looped back to the input of the Batcher network, and the Batcher-banyan network is again used to acknowledge the input ports. Note that, in this phase, the source address is used by the sorting and the self-routing mechanism for sending the acknowledgment packets. The input ports receiving the acknowledgment for their request then transmit the full packet through the same Batcher-banyan network. This process constitutes the third and final phase. Obviously, there will be no conflict at the output ports in this phase. Input ports which fail to receive an acknowledgment retain the packet in their input queue and retry in the next cycle when the three-phase operation is repeated again. The queueing of the unacknowledged packet at the head of the queue will cause the subsequent packets which might have been intended for a free output port to be delayed. This phenomenon is called head of the line (HOL) blocking, and in principle will reduce the switch throughput.

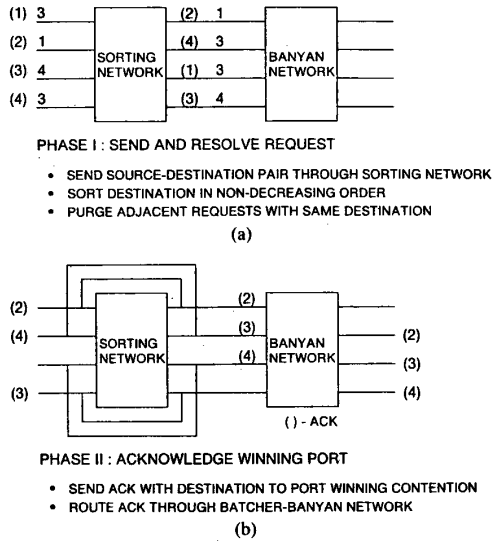


Fig. 7. Phases I and II in the switch proposed in [42], [43]. From [43], © 1987 IEEE.

The switch supports a port speed of 150 Mbits/s. Of course, phases one and two constitute overhead processing for the switch fabric. Therefore, the switch fabric would have to be speeded up by a fraction which depends on the size of the switch fabric and the size of the information field of the packet. For example, as shown in [43], for a 1000 × 1000 switch and the packet size of 1000 bits, the overhead is about 14 percent. This means that the switch has to operate at 170 Mbits/s in order to handle a 150 Mbit/s input port speed.

The performance study of this switch for random traffic indicates that the maximum throughput is about 58 percent. Actually, this is a theoretical limit which can be shown for any nonblocking space switch operating in a packet-switched mode which employs input queueing with

random traffic (see [44]). Results in [43] also indicate that, with a reasonable input buffer size, an acceptable buffer overflow probability can be achieved.

The performance behavior and the maximum throughput of this switch would be different if the traffic behavior were nonrandom and specifically periodic. In [43], there is a discussion on how to improve the throughput of the switch in the presence of periodic traffic. In a theoretical paper, Li [45] studies the performance of a generic nonblocking packet switch with input queues and periodic packet traffic streams. He determines the necessary and sufficient switch-internal clock rate as a function of the input stream rate in order for the switch to be nonblocking.

The use of a Batcher-banyan network was also proposed in the *Synchronous Wideband Switch* in [46], [47] which is a time–space–time circuit switching system. The function of the Batcher-banyan fabric is to provide a nonblocking self-routing network in the space segment of the time–space–time switch. The time slot interchangers at the inputs and the outputs of the Batcher-banyan network rearrange the time slots to avoid port contention.

The technological feasibility of the Batcher-banyan concept is described in [33]. A Batcher chip of size 32 × 32 has been built in CMOS technology with each port running at 140 Mbits/s. Furthermore, about 100 of those chips are packaged in a three-dimensional way to form a 256 × 256 Batcher-banyan switch fabric with a total throughput of 35 Gbits/s.

Fabrics with Disjoint-Path Topology and Output Queueing

The switch fabric architectures described so far were based on multistage interconnection networks comprised of small switching elements. The switch fabrics described in this section are based on a fully interconnected topology in the sense that every input has a nonoverlapping direct path to every output so that no blocking or contention may occur internally. They employ output queueing in order to resolve the output port contention. Intuitively speaking, in any nonblocking space-division packet switch, a higher throughput can be achieved with output queueing as compared to input queueing. This is because the head of the line blocking effect which is the limiting factor in a switch with input queues disappears with output queueing. This result has been shown analytically by Karol and Hluchyj in [44], [49]. In fact, one can show that a nonblocking space-division switch with an output queueing of infinite capacity would give the best delay/throughput performance.

The first switch fabric we describe in this class is the *Knockout* switch [50], [51]. The Knockout switch is designed for a pure packet-switched environment and can handle either fixed-length (called Knockout I [50]) or variable-length packets (called Knockout II [51]). The Knockout switch uses one broadcast input bus from every input port to all output ports as shown in Fig. 8. Each output port has a bus interface which can receive packets

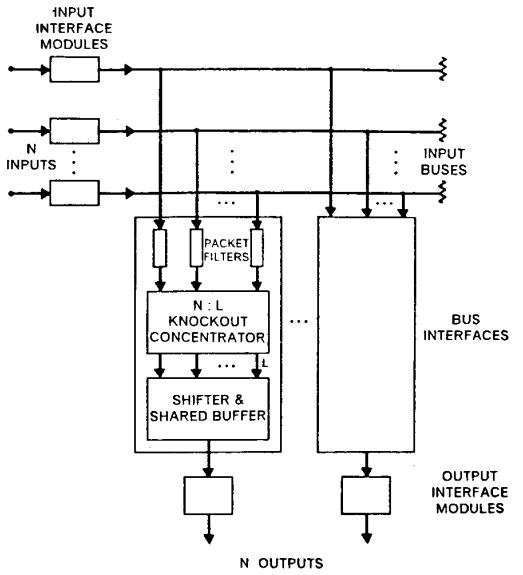


Fig. 8. Basic structure of the Knockout switch. From [75], © 1988 IEEE.

from each input bus line or input port. Hence, no contention occurs between packets destined for different outputs. In addition, simultaneous packets from several inputs can be transmitted to the same output. In Fig. 8, one of the output bus interfaces is shown in more detail. It has three major components. The first component is the set of N packet filters, each interfacing a bus line. The packet filters, which implement the self-routing function, detect the address of each packet on the broadcast bus and pass those destined to that output on to the next component which is the concentrator. The concentrator uses a novel algorithm to select a fixed number of packets, say L, from the N incoming lines to the concentrator. The L selected packets are stored in the order of their arrivals into a shared buffer which constitutes the third component. The main philosophy behind the N to L concentration mechanism is that the probability of packet loss due to output congestion can be kept below the loss expected from other sources such as channel errors. This means, for example, that with N large and $L = 8$, a packet loss rate of less than 10^{-6} can be achieved. Taking advantage of this observation, the number of separate buffers needed to receive simultaneously arriving packets is reduced from N to L. This result holds under the assumption of uniform traffic patterns. If the traffic pattern is nonuniform, L has to be higher (e.g., up to 20) for the same packet loss rate, depending on the nonuniformity of traffic [52].

The basic principle of selecting L packets out of N possible contenders in the concentrator stage is an algorithm implemented in the hardware analogous to a knockout tournament. For a concentrator with N inputs and L outputs, there are L rounds of competition. The basic building block of the concentrator is a 2 × 2 contention switching element, with one output being the winner and one the loser. When two packets arrive, one is selected ran-

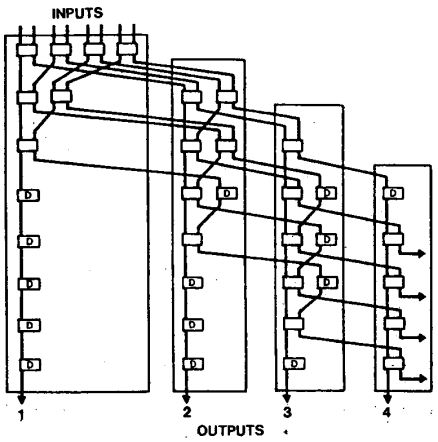


Fig. 9. Example for an 8:4 Knockout concentrator. From [50], © 1987 IEEE.

domly as a winner. Fig. 9 shows a block diagram of an 8:4 concentrator made of these 2 × 2 contention switches. The boxes marked "*D*" indicate a 1-bit delay line to keep the competition synchronous. The first round of the tournament starts with N contenders. The $N/2$ winners from the first round advance to the second round. The winners in the second round advance to the third round, and so on. Note that the final winner at the output number one has won all rounds, the winner at output number two has won all but one round, and so on.

The next segment after the concentrator is a shared buffer structure. The output buffer must be capable of storing up to L packets within one packet time slot. This is achieved by using L separate buffers preceded by a shifter function. This logically implements an L-input single-output FIFO queueing discipline for all packets arriving at the output.

The design of the Knockout switch is based on possible VLSI realization with input/output and internal hardware operating at 50 Mbits/s. Also, a solution is proposed for modular growth which can grow from 32 × 32 to 1024 × 1024.

Another switch fabric, the design of which is based on an interconnection structure with no internal blocking and queueing capability at the outputs of it modules, is the *Integrated Switch Fabric* proposed in [53]. This switch fabric is designed to handle both circuit-switched and packet-switched traffic in a unified manner. It is self-routing, and uses uniform fixed-length *minipackets* within the switching fabric for all types of connections. Circuit-switched connections can be provided for various speeds and for constant delay with full transparency to the terminating ports. Its design concept emphasizes modularity; it is based on VLSI technology, aiming for a single chip as the basic building block such that a very large range of switch fabrics can be configured, covering sizes from 16 to more than 1000 input/output ports. This translates into a throughput capability from 500 Mbits/s to 30 Gbits/s, assuming a speed of 32 Mbits/s per port.

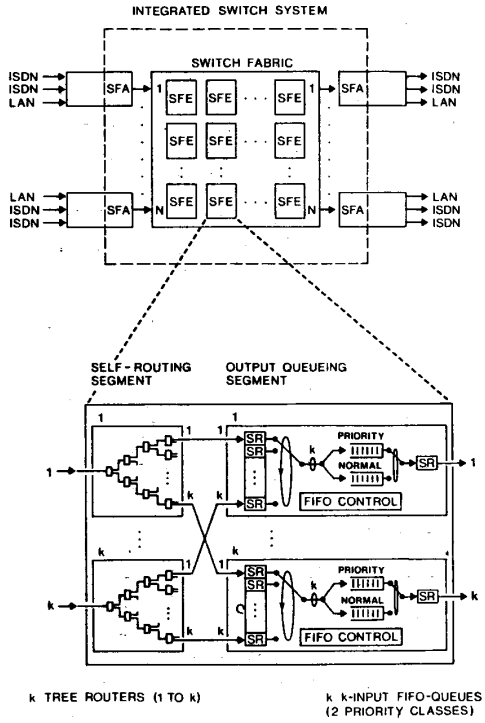


Fig. 10. Basic switch structure and switch fabric element proposed in [53].

Fig. 10 shows the basic structure of the fabric which consists of two major components: the switch fabric adapters (SFA) and the switch fabric elements (SFE). The function of the switch fabric adapter is to convert the user information from packet-switching and circuit-switching interfaces into uniform fixed-length minipackets. Fig. 10 also shows the basic structure of a $k \times k$ switch fabric element which consists of a self-routing segment of an output queueing segment. The self-routing segment of an SFE performs the minipacket routing function. It is a decoder with a tree structure per input which can simultaneously route minipackets from every input to every output. Each node of the tree decoder segment has one input and two outputs and works like a simple banyan node with only one used input. Therefore, within each module, up to k minipackets can be transferred to the same output at the same time. The interface between the terminating points of all trees belonging to a certain SFE output and the output line consists of k small shift registers (SR) for intermediate storage of minipackets and a pair of FIFO queues which constitute the output queueing segment. The importance of the shift registers is to store the minipackets momentarily to allow sequential access to the output FIFO queue pair. The contents of the shift registers are transferred sequentially into the associated output FIFO at a speed k times higher which is realized by parallelism. The combination of shift registers and their corresponding output FIFO is a realization of a multiinput single-output FIFO queue. One major difference between this switch

fabric and most others is that it uses a priority scheme for different classes of traffic. Two priority classes are proposed. One is used for the high-priority or time-critical type traffic, and the other for the low-priority traffic. Circuit-switched connections are supported with minipackets which have high priority. Another feature of this switch architecture is its simple modular growth capability. That is, larger switch sizes can be configured by combining $k \times k$ switch fabric elements via stage-expanding and/or multistage arrangements.

A single-stage $N \times N$ switch fabric can be built from the basic $k \times k$ element. Fig. 11 shows an example of a $2k \times 2k$ switch fabric configured from four $k \times k$ switch fabric elements. Each basic module has a selection logic which can be set such that only the appropriate module accepts the incoming minipackets and routes them internally to the destined output. Furthermore, in each module, there are provisions for each output to ensure that only one output FIFO queue is feeding the output line at a time. In the same manner, a $4k \times 4k$ fabric can be realized from a $2k \times 2k$ fabric, and so on. In general, to build a single-stage $N \times N$ fabric from a basic $k \times k$ element, $(N/k)^2$ modules are required. It should be noted that a single-stage switch fabric which is realized this way still preserves its disjoint-path topology with output queueing. However, each output of the $N \times N$ single-stage configuration has one logical queueing segment which is physically realized in different switch fabric elements as parallel queues. For a moderate-size switch, say up to 128×128, the single-stage approach seems feasible and reasonable. But, for a large-size switch, a three-stage realization is proposed which yields a very cost-effective solution.

Another multistage switch fabric architecture which actually considers the buffered banyan structure but with larger modules has been proposed in [54]–[56]. We have included it in this section because each module or switching element has a structure with disjoint-path topology and a substantial output buffering at each of the module outputs. The switching element has eight inputs and eight outputs. A TDM bus interconnecting all input ports to all output ports runs at the sum of all input bit rates and hence is nonblocking. The switch runs at 560 Mbits/s per port. The bus speed within each module is therefore 8×560 Mbits/s. The output queues of each module can be filled at the speed of the TDM bus.

The overall network [56] functionally constitutes a randomization (distribution) network preceding a banyan routing network which actually maps into a Benes topology. In reality, the randomization network and routing network are folded and combined into one network with input/output pairs at one side and a mirror line at the other side of the network. Hence, it is used in one direction for the randomization function and in the other direction for the routing function. Reference [56] presents simulation results of this switch with emphasis on video application.

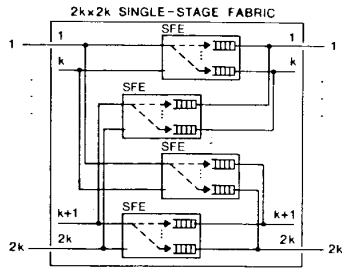


Fig. 11. Single-stage expansion in the switch proposed in [53], © 1988 IEEE.

Crossbar-Based Switch Fabrics

Crossbar switches have always been attractive to switch designers because they are internally nonblocking and they are simple. In addition to circuit-switch applications, they have also been considered as a base for switches which operate in a packet mode. But unfortunately, the simple crossbar matrix has the property of square growth and is not economical for large switches. Nearly all known approaches are either designed for relatively small applications or just for a building block which is used in larger multistage arrangements.

Even though the crossbar matrix is internally nonblocking, in packet mode operation, the probability of output port contention remains. Hence, a queueing function has to be added to the pure crosspoint matrix in order to overcome that problem. The location of this queueing function allows the approaches of crossbar-based switches to be categorized. In principle, there are three possibilities, namely, crossbar matrices with input queueing, matrices with queueing within the crosspoints themselves, and matrices with output queueing.

If the queueing function is located at the inputs of the crossbar matrix, then a switch control is necessary which arbitrates all packets waiting at the heads of the different input queues and which are destined for a certain output port.

One solution is a centralized control. This control gets requests over separate control paths from all switch input ports which have packets waiting. It schedules these requests, sets up the necessary crosspoints in the matrix, and grants the requests as soon as the packets can be transmitted. This way, the matrix itself remains as simple as possible and can be realized economically with high-speed technology. By use of several matrices in parallel, even higher speeds can be achieved. Certainly, the central control is the bottleneck of this approach. The control overhead drastically increases with the size of the switch. Nevertheless, with suitable technology and intelligent techniques (e.g., pipelining), very high-speed, centrally controlled packet switches can be built, as long as they are small.

One way to reduce the control overhead is to distribute the control function. This leads to an approach as pro-

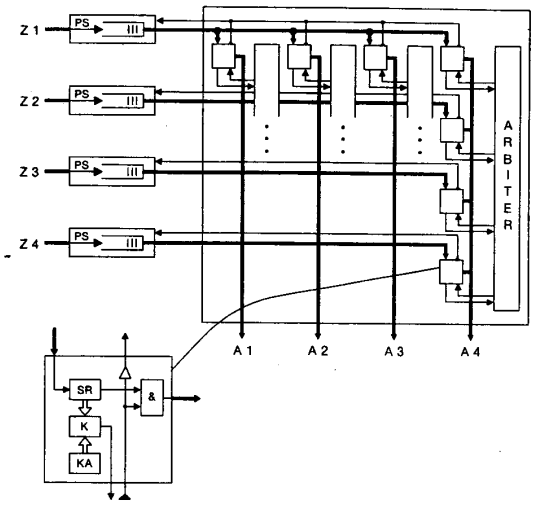


Fig. 12. The crossbar switch with input queueing proposed in [57], [58]. From [57], © 1987 vde-verlag gmbh.

posed, e.g., in [57], [58]; see Fig. 12. There, each output port has its individual control, called arbiter. This arbiter allows only one of the input ports to be switched through at the same time to the corresponding output port by a fair algorithm. This is accomplished by a separate control signal (back pressure) to each input queue which practically stops all but one of those queues which compete for the same output port. Since only one of the arbiters generates a back-pressure signal to a specific input queue at the same time, all of these signals of a corresponding matrix line can be interconnected very simply by a wired OR connection. In addition, this switch proposal does not need separate request lines from the input queues to the arbiters because it has the address decoder function distributed in the crosspoints of the matrix. Hence, this approach represents a self-routing crossbar switch.

The performance of crossbar matrices with input queueing depends very much on the control overhead, and especially on the way in which the input queues are organized. If the input queue is a simple, single FIFO queue, then the throughput of the switch saturates even at 58 percent (see [44], [49]) or less, depending on the control overhead and the traffic distribution. This is due to the head of the line (HOL) blocking phenomenon, already mentioned earlier. This means that the waiting packet at the head of the queue eventually blocks subsequent packets which might be destined for momentarily free output ports. But, if the input queue is organized as multiple queues, i.e., one per destination port, then we come closer to the concept of output queueing, even though these queues are physically located at the inputs. As mentioned earlier, output queueing provides ideal performance. The extent to which the performance of input queueing with multiple queues per input measures up to that of output queueing depends largely on how the input queues are controlled and scheduled and what the control overhead is.

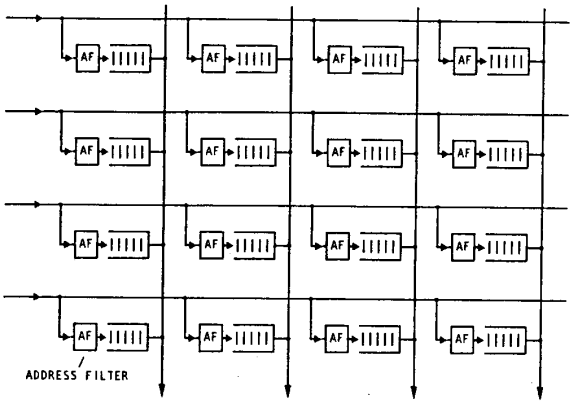


Fig. 13. A crossbar switch with queueing in the crosspoints.

A second class of crossbar switches has been proposed which implements the queueing function within the crosspoints themselves. The simple on/off switch of a crosspoint is replaced now by a FIFO queue preceded by an address decoder function (packet filter). So, the packets can be sent directly into the matrix. A packet can only pass through the filter whose address matches the packet's destination address. So, the self-routing concept is realized. The queues of a matrix column which belong to one specific output port have to be emptied in a fair way, e.g., in round-robin fashion. This is also done by one very simple arbiter per output port.

The concept is shown in Fig. 13. It has been proposed, e.g., in [59], in the *Bus Matrix Switch* described in [60], or in the experimental broadband ATM switching system described in [61]. The architecture is motivated by the fact that the concept provides almost ideal performance, at least as long as the queues are dimensioned long enough. On the other side, because of the square growth, it is a very expensive approach for large matrices and can only be a good solution either for small switches or as basic architecture for the building blocks of modular multistage arrangements, as proposed in [60] and [61].

Finally, looking at the third class of crossbar switches, namely, matrices with output queueing, it has to be mentioned that they only work if the matrix runs at N times higher speed, assuming a matrix size of $N \times N$ (see [49]). Under this assumption, they are not attractive for very high-speed switches. So, parallelism has to be introduced instead of speed. This leads to an expanded $N \times N^2$ matrix, and finally to the class of switches with disjoint-path topology and output queueing which we already covered separately in a previous section of this paper.

Time Division Fabrics with Common Packet Memory

A switch architecture which uses a common memory for all connection paths is the *Prelude* switch [62]–[66];

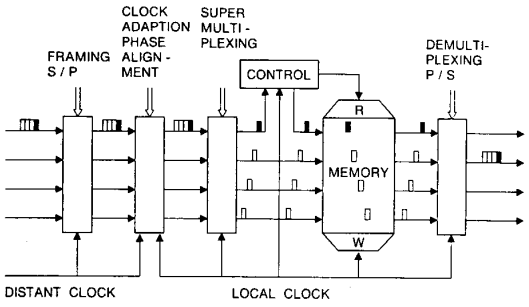


Fig. 14. Basic structure of the Prelude switch. From [66], © 1987 IEEE.

see Fig. 14. Even though this system works in a packet mode, it operates very similarly to a conventional TDM circuit switch. This mode is called asynchronous time division (ATD) and is characterized by a slotted operation with packet queueing. The incoming packets, one per frame, are synchronized first and converted from serial to parallel so they can be written cyclically to the common memory. In advance, the control information is extracted from the packet headers and fed to the common control. Since the control has to handle all packets sequentially, the multiplexing at the input is done such that the headers of all packets arrive at the control sequentially. The control compares the header information with a routing table which delivers a new header for the packet. Additionally, a control memory provides the addressing information for the memory read process. Unlike a TDM switch where the control memory is operated cyclically, here it is operated as FIFO queues. Finally, the outgoing information is demultiplexed again and converted back from parallel to serial.

It has been demonstrated in a Prelude prototype [62] that a module of size 16 × 16 can be built for a line speed of 280 Mbits/s with a reasonable amount of hardware. This is due to the use of high internal parallelism and fast technology (ECL) at the same time. Since the basic architecture is not suited for large switches, the Prelude authors also suggest that large switches be constructed in a modular way by interconnecting smaller building blocks.

Fabrics with Shared Medium

In this section, we classify switch fabric architectures which are based on a shared medium as switching kernel. We concentrate especially on fast packet switch approaches which employ a bus or ring network as switching medium. But we do not want to cover here all the typical random access local-area network architectures.

Ring and bus networks are used as switching media in many of today's packet switches. Their technology is well understood and advanced. They provide flexibility in terms of the access protocol and distribution of traffic. However, their bandwidth capacity and throughput are limited compared to multipath switch architectures. One way to increase the capacity limitation is to use multiple

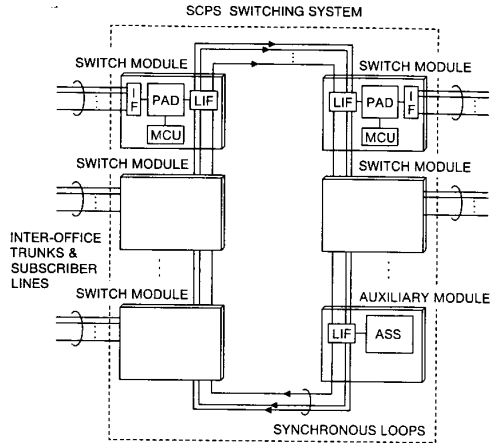


Fig. 15. Basic structure of the SCPS switch. From [70], © 1987 IEEE.

rings or multiple buses in a single or multiple hierarchical structure.

An experimental *Synchronous Composite Packet Switching* (SCPS) architecture [67]–[70] was proposed for integrated circuit- and packet-switching functions which uses multiple rings to interconnect the switching modules. The switching system architecture as shown in Fig. 15 comprises switch modules interfacing externally to circuit-switched and packet-switched lines. The switch modules are interconnected by multiple fiber optical rings where each module has a direct interface to each ring. The switching operations are accomplished by transmitting the circuit- and packet-switched channel messages between switch modules via the intermodule network. Intermodule signaling and command messages for system operation and control are transmitted between the modules as well.

The circuit, packet, and signaling information are all transmitted within the switching subsystem in similarly structured packets which are generated in the following way. Each switch module generates a so-called *composite packet* from several circuit-switched channel samples which have to be transmitted simultaneously and have the same destination module. Conventional data user packets are converted into switch packets, called *noncomposite packets*, in which the original packet with its header is encapsulated by an SCPS header and trailer. The signaling and control packets have similar headers and trailers as the other two types of packets. Thus, all three kinds of packets have almost the same structure.

The intermodule network consists of 8 (or 16) independent synchronous optical loops with loop speeds of 393 Mbits/s (98.3 Mbits/s), connecting 32 modules. The SCPS system has a total throughput of 4 Gbits/s (2 Gbits/s). Note that the intramodule switching is done locally; hence, the maximum system throughput is higher

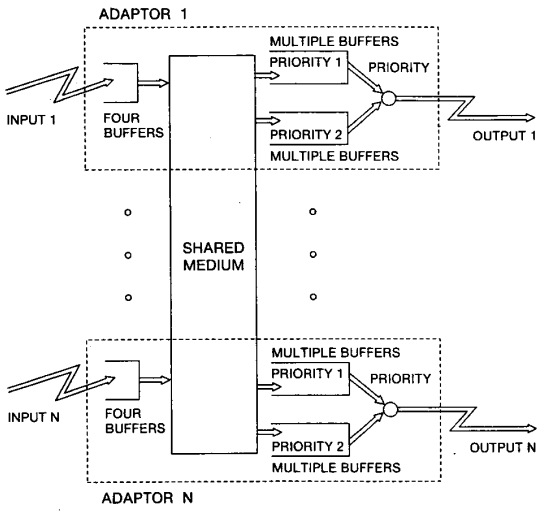


Fig. 16. Basic structure of the PARIS switch.

than the maximum throughput supported by multiple loops. The access method to loops is similar to the slotted ring method [70]. The individual loops operate synchronously. Having a frame structure with a duration of 125 μs, complete time transparency for circuit-switched channels is maintained. Each frame consists of 384 (96) time slots of 128 bits each. This switch architecture can integrate $n \times 64$ kbit/s circuit switching with packet switching in any combination while maintaining its compatibility with existing networks. By using composite packets for circuit-switched channels, the delay for packet assembling and disassembling can be reduced to a value comparable to the conventional TDM switching systems.

A similar switch architecture which also uses multiple rings to provide the fast-packet switching function was also reported in [71].

Several switch architectures have used a shared bus structure as their transport or switching mechanism (see, e.g., [16]). In fact, the switch fabric proposed in [16] is one of the very early switching concepts that used the concept of packet switching as the only means to transport a wide range of services such as voice, data, and video.

A recent experimental high-speed packet switching system which was designed to transport voice, data, and video all in packetized form and uses a bus architecture is the *Packetized Automated Routing Integrated System* (PARIS) [72]. Its design philosophy is to use a very simple protocol to achieve low packet delay with an architecture simple enough that it can be implemented even with off-the-shelf components.

The basic structure of the PARIS switch is shown in Fig. 16. It uses a high-speed bus as shared medium to interconnect input ports to the output ports [73]. The maximum bandwidth of the shared bus is taken to be greater than the aggregate capacity of all input lines. The switch can handle variably sized packets ranging from 32 bits to a maximum of 8 kbits. Each input port has a buffer that can hold no more than four packets of maximum size. It is shown that this buffering is adequate to ensure no packet loss, provided a round-robin exhaustive service policy is used to arbitrate the access to the bus. Therefore, the switch is nonblocking. The arbitration protocol is implemented using a fast token-passing algorithm [73]. Each output port has two buffers, one for each priority class. The output buffers are sized such that the packet loss due to a momentary overload situation of an output port is less than 10^{-8}.

Summary

We have presented an overview of the major high-performance telecommunications switching fabrics which have been proposed and experimentally developed within the past few years. Most of the architectures we covered in this paper use the principle of fast packet switching as a unified switching architecture for the transport of a wide range of services with different bandwidth requirements. Our intention was to give a descriptive overview by attempting to classify the approaches into six categories. The classification has been chosen in order to emphasize the basic principles which differentiate these architectures.

Some of these basic principles inherently allow covering a wide range of fabric sizes while maintaining their basic architecture in a clean way. Other basic principles are limited, such that they can only be considered for small fabrics. Some principles are also applicable as the basic architecture for modules in large multistage fabrics. The way modules are interconnected could be viewed as a basic principle itself.

Considering only small fabrics, one can make the general observation that almost every known principle allows building small fabrics of approximately the same capabilities and performance. Large multistage fabrics, however, are more sensitive to the architecture of their modules. Several module designs, if they are considered isolated from the whole fabric, may show only minor differences in their module performance. But, when placed within a multistage environment, these minor differences could escalate drastically. In general, multistage fabrics perform better if they employ queueing within the stages or the switch modules, respectively. Furthermore, the location of the queues within the modules is important [38]. Hence, it seems to be worth directing some effort to optimizing the design of a buffered, self-routing, large-fabric switch module.

Finally, we have to mention that in our overview, we only concentrated on the point-to-point connection aspects of these switching architectures. Many of the switching architectures described here have also been enhanced to support broadcasting and multicasting connections using fast packet-switching principles (see, e.g., [29], [74], [75]) which we have not covered other than just mentioning them.

REFERENCES

[1] H. Frank *et al.*, "Issues in the design of networks with integrated voice and data" in *Proc. ICC'77*, June 1977, pp. 38.1.36-38.1.43.

[2] M. J. Ross *et al.*, "Design approaches and performance criteria for integrated voice/data switching," *Proc. IEEE*, vol. 65, pp. 1283-1295, Sept. 1977.

[3] H. Frank and I. Gitman, "Study shows packet switching is best for voice traffic, too," *Data Commun.*, pp. 43-62, Mar. 1979.

[4] I. Gitman and H. Frank, "Economic analysis of integrated voice and data networks: A case study," *Proc. IEEE*, vol. 66, pp. 1549-1570, Nov. 1978.

[5] H. Frank, "Plan today for tomorrow data/voice nets," *Data Commun.*, pp. 51-62, Sept. 1978.

[6] J. S. Turner and L. F. Wyatt, "A packet network architecture for integrated services," in *Proc. GLOBECOM'83*, San Diego, CA, Nov. 1983, pp. 2.1.1-2.1.6.

[7] J. S. Turner, "New directions in communications (or which way to the information age?)," *IEEE Commun. Mag.*, vol. 24, pp. 8-15, Oct. 1986. Also, in *Proc. Int. Zurich Seminar Digital Commun.*, Zurich, Switzerland, Mar. 1986, pp. A3.1-A3.8.

[8] J. J. Kulzer and W. A. Montgomery, "Statistical switching architecture for future services," in *Proc. ISS'84*, Florence, Italy, May 1984, pp. 43A.1.1-43A.1.6.

[9] R. E. Cardwell and J. H. Campbell, "Packet switching of data with the 3B-20D computer system," in *Proc. ISS'84*, Florence, Italy, May 1984, pp. 22A.4.1.-22A.4.6.

[10] J. F. Huber and E. Mair, "A flexible architecture for small and large packet switching networks," in *Proc. ISS'87*, Phoenix, AZ, Mar. 1987, pp. B10.4.1-B10.4.6.

[11] M. J. Ross *et al.*, "An architecture for a flexible integrated voice/data network," in *Proc. ICC'80*, June 1980, pp. 21.6.1-21.6.5.

[12] J. D. Morse and S. J. Kopec, Jr., "Performance evaluation of a distributed burst-switching communications system," in *Proc. Phoenix Conf. Comput. Commun.*, Mar. 1983.

[13] S. R. Amstutz, "Burst switching—A method for dispersed and integrated voice and data switching," *IEEE Commun. Mag.*, vol. 21, pp. 36-42, Nov. 1983. Also, in *Proc. ICC'83*, June 1983, pp. 288-292.

[14] E. F. Haselton, "A PCM frame switching concept leading to burst switching network architecture," *IEEE Commun. Mag.*, vol. 21, pp. 13-19, Sept. 1983. Also, in *Proc. ICC'83*, Boston, MA, June 1983, pp. 1401-1406.

[15] R. Rettberg *et al.*, "Development of voice funnel system: Design report," Bolt, Beranek and Newman Inc., Rep. 4098, Aug. 1979.

[16] E. H. Rothauser, P. A. Janson, and H. R. Mueller, "Meshed-star networks for local communication systems," in *Local Networks for Computer Communications*, A. West and P. Janson, Eds. Amsterdam: North-Holland, 1981, pp. 25-41.

[17] C. Clos, "A study of non-blocking switching network," *Bell Syst. Tech. J.*, vol. 32, pp. 406-424, Mar. 1953.

[18] V. E. Benes, "Optimal rearrangeable multistage connecting networks," *Bell Syst. Tech. J.*, vol. 43, pp. 1641-1656, July 1964.

[19] L. R. Goke and G. J. Lipovski, "Banyan networks for partitioning multiprocessor systems," in *Proc. 1st Annu. Int. Symp. Comput. Architecture*, Dec. 1973, pp. 21-28.

[20] R. J. McMillan, "A survey of interconnection networks," in *Proc. GLOBECOM'84*, Atlanta, GA, Dec. 1984, pp. 105-113.

[21] J. H. Patel, "Processor-memory interconnections for multiprocessors," in *Proc. 6th Annu. Int. Symp. Comput. Architecture*, Apr. 1979, pp. 168-177.

[22] ——, "Performance of processor-memory interconnections for multiprocessors," *IEEE Trans. Comput.*, vol. C-30, pp. 771-780, Oct. 1981.

[23] D. M. Dias and J. R. Jump, "Analysis and simulation of buffered delta networks," *IEEE Trans. Comput.*, vol. C-30, pp. 273-282, Apr. 1981.

[24] ——, "Packet switching interconnection networks for modular systems," *IEEE Comput. Mag.*, vol. 14, pp. 43-53, Dec. 1981.

[25] D. M. Dias and M. Kumar, "Packet switching in $N \log N$ multistage networks," in *Proc. GLOBECOM'84*, Atlanta, GA, Dec. 1984, pp. 114-120.

[26] C. P. Kruskal and M. Snir, "The performance of multistage interconnection networks for multiprocessors," *IEEE Trans. Comput.*, vol. C-32, pp. 1091-1098, Dec. 1983.

[27] M. Kumar and J. R. Jump, "Performance of unbuffered shuffle-exchange networks," *IEEE Trans. Comput.*, vol. C-35, pp. 573-577, June 1986.

[28] J. S. Turner, "Fast packet switch," U. S. Patent 4 491 945, Jan. 1, 1985.

[29] ——, "Design of an integrated services packet network," in *Proc. 9th ACM Data Commun. Symp.*, Sept. 1985, pp. 124-133.

[30] P. Kermani and L. Kleinrock, "Virtual cut-through: A new computer communication switching technique," *Comput. Networks*, vol. 3, pp. 267-286, 1979.

[31] R. W. Muise, T. J. Shonfeld, and G. H. Zimmerman, "Digital communications experiments in wideband packet technology," in *Proc. Int. Zurich Seminar Digital Commun.*, Zurich, Switzerland, Mar. 1986, pp. 135-140.

[32] G. W. R. Luderer *et al.*, "Wideband packet technology for switching systems," in *Proc. ISS'87*, Phoenix, AZ, Mar. 1987, pp. 448-454.

[33] A. K. Vaidya and M. A. Pashan, "Technology advances in wideband packet switching," in *Proc. GLOBECOM'88*, Hollywood, FL, Nov. 1988, pp. 668-671.

[34] J. S. Turner, "Design of a broadcast packet switching network," in *Proc. INFOCOM'86*, Apr. 1986, pp. 667-675. Also, Dep. Comput. Sci., Washington Univ., St. Louis, MO, Tech. Rep. WUCS-84-4, Mar. 1985.

[35] Y.-C. Jenq, "Performance analysis of a packet switch based on a single-buffered banyan network," *IEEE J. Select. Areas Commun.*, vol. SAC-1, pp. 1014-1021, Dec. 1983.

[36] R. G. Bubenik and J. S. Turner, "Performance of a broadcast packet switch," Dep. Comput. Sci., Washington Univ., St. Louis, MO, Tech. Rep. WUCS-86-10, Mar. 1986.

[37] M. N. Huber, E. P. Rathgeb, and T. H. Theimer, "Self routing banyan networks in an ATM-environment," in *Proc. ICCC'88*, Tel Aviv, Israel, Oct. 1988, pp. 167-174.

[38] E. P. Rathgeb, T. H. Theimer, and M. N. Huber, "Buffering concepts for ATM switching networks," in *Proc. GLOBECOM'88*, Hollywood, FL, Nov. 1988, pp. 1277-1281.

[39] K. E. Batcher, "Sorting networks and their application," in *Proc. Spring Joint Comput. Conf.*, AFIPS, 1968, pp. 307-314.

[40] A. Huang and S. Knauer, "Starlite: A wideband digital switch," in *Proc. GLOBECOM'84*, Atlanta, GA, Dec. 1984, pp. 121-125.

[41] A. Huang, "The relationship between STARLITE, A wideband digital switch and optics," in *Proc. ICC'86*, Toronto, Canada, June 1986, pp. 1725-1729.

[42] J. Y. Hui, "A broadband packet switch for multi-rate services," in *Proc. ICC'87*, Seattle, WA, June 1987, pp. 782-788.

[43] J. Y. Hui and E. Arthurs, "A broadband packet switch for integrated transport," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 1264-1273, Oct. 1987.

[44] M. J. Karol, M. G. Hluchyj, and S. P. Morgan, "Input versus output queueing on a space-division packet switch," *IEEE Trans. Commun.*, vol. COM-35, pp. 1347-1356, Dec. 1987.

[45] S.-Y. R. Li, "Theory of periodic contention and its application to packet switching," in *Proc. INFOCOM'88*, New Orleans, LA, Mar. 1988, pp. 320-325.

[46] L. T. Wu and N. C. Huang, "Synchronous wideband network—An interoffice facility hubbing network," in *Proc. Int. Zurich Seminar Digital Commun.*, Zurich, Switzerland, Mar. 1986, pp. 33-39.

[47] C. M. Day, J. N. Giacopelli, and J. Hickey, "Applications of self-routing switching to LATA fiber optic networks," in *Proc. ISS'87*, Phoenix, AZ, Mar. 1987, pp. 519-523.

[48] W. D. Sincoskie, "Frontiers in switching technology; Part two: Broadband packet switching," *Bellcore Exchange*, pp. 22-27, Nov./Dec. 1987.

[49] M. G. Hluchyj and M. Karol, "Queueing in space-division packet switching," in *Proc. INFOCOM'88*, New Orleans, LA, Mar. 1988, pp. 334-343.

[50] Y. S. Yeh, M. G. Hluchyj, and A. S. Acampora, "The Knockout switch: A simple, modular architecture for high-performance packet switching," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 1274-1283, Oct. 1987.

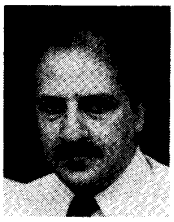
[51] K. Y. Eng. M. G. Hluchyj, and Y. S. Yeh, "A Knockout switch for variable-length packets," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 1426-1435, Dec. 1987.

[52] H. Yoon, M. T. Liu, and K. Y. Lee, "The Knockout switch under nonuniform traffic," in *Proc. GLOBECOM'88*, Hollywood, FL, Nov. 1988, pp. 1628-1634.

[53] H. Ahmadi *et al.*, "A high-performance switch fabric for integrated circuit and packet switching," in *Proc. INFOCOM'88*, New Orleans, LA, Mar. 1988, pp. 9-18.

[54] P. Debuysscher, J. Bauwens, and M. De Somer, "Système de commutation," Belgian Patent BE 904100, Jan. 1986.

[55] M. De Prycker and J. Bauwens, "A switching exchange for an asynchronous time division based network," in *Proc. ICC'87*, Seattle, WA, June 1987, pp. 774-781.
[56] M. De Prycker and M. De Somer, "Performance of a service independent switching network with distributed control," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 1293-1301, Oct. 1987.
[57] U. Killat, "Asynchrone Zeitvielfachübermittlung für Breitbandnetze," *Nachrichtentech. Z.*, vol. 40, no. 8, pp. 572-577, 1987.
[58] W. Jasmer, U. Killat, and J. Krüger, German Patent Appl. P 37 14 385.9.
[59] R. Bakka and M. Dieudonne, "Switching circuit for digital packet switching network," U.S. Patent 4 314 367, Feb. 2, 1982.
[60] S. Nojima *et al.*, "Integrated services packet network using bus matrix switch," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 1284-1292, Oct. 1987.
[61] Y. Kato *et al.*, "Experimental broadband ATM switching system," in *Proc. GLOBECOM'88*, Hollywood, FL, Nov. 1988, pp. 1288-1292.
[62] A. Thomas, J. P. Coudreuse, and M. Servel, "Asynchronous time division techniques: An experimental packet network integrating video communication," in *Proc. ISS'84*, Florence, Italy, May 1984, paper 32C2.
[63] P. Gonet, "Fast packet approach to integrated broadband networks," *Comput. Commun.*, pp. 292-298, Dec 1986.
[64] P. Gonet, P. Adams, and J. P. Coudreuse, "Asynchronous time-division switching: The way to flexible broadband communication networks," in *Proc. Int. Zurich Seminar Digital Commun.*, Zurich, Switzerland, Mar. 1986, pp. 141-148.
[65] M. Dieudonne and M. Quinquis, "Switching techniques for asynchronous time division multiplexing (or fast packet switching)," in *Proc. ISS'87*, Phoenix, AZ, Mar. 1987, pp. 367-371.
[66] J. P. Coudreuse and M. Servel, "Prelude: An asynchronous time-division switched network," in *Proc. ICC'87*, Seattle, WA, June 1987, pp. 769-773.
[67] T. Takeuchi and T. Yamaguchi, "Synchronous composite packet switching for ISDN switching system architecture," in *Proc. ISS'84*, Florence, Italy, May 1984, p 42B3.
[68] T. Takeuchi *et al.*, "An experimental synchronous composite packet switching system," in *Proc. Int. Zurich Seminar Digital Commun.*, Zurich, Switzerland, Mar. 1986, pp. 149-153.
[69] H. Suzuki, T. Takeuchi, and T. Yamaguchi, "Very high speed and high capacity packet switching for broadband ISDN," in *Proc. ICC'86*, Toronto, Canada, June 1986, pp. 749-754.
[70] T. Takeuchi *et al.*, "Synchronous composite packet switching-A switching architecture for broadband ISDN," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 1365-1376, Oct. 1987.
[71] J. R. Pierce, "Network for block switching of data," *Bell Syst. Tech. J.*, July-Aug. 1972.
[72] I. S. Gopal, I. Cidon, and H. Meleis, "Paris: An approach to integrated private networks," in *Proc. ICC'87*, Seattle, WA, June 1987, pp. 764-773.
[73] I. Cidon, I. S. Gopal, and S. Kutten, "New models and algorithms for future networks," in *Proc. ACM Principles Distributed Comput.*, Toronto, Canada, 1988.
[74] T. T. Lee, R. Boorstyne, and E. Arthurs, "The architecture of a multicast broadband packet switch," in *Proc. INFOCOM'88*, New Orleans, LA, Mar. 1988, pp. 1-8.
[75] K. Y. Eng, M. G. Hluchyj, and Y. S. Yeh, "Multicast and broadcast services in a knockout packet switch," in *Proc. INFOCOM'88*, New Orleans, LA, Mar. 1988, pp. 29-34.



Hamid Ahmadi (S'78-M'83) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Columbia University, New York, NY, in 1976, 1978, and 1983, respectively.

From 1980 to 1984 he was with the Switching and Signaling Systems Department at Bell Laboratories, Holmdel, NJ, where he was involved in the area of performance analysis and protocol studies of signaling networks. He joined the IBM T. J. Watson Research Center, Yorktown Heights, NY, in 1984, and has since been a member of the Telecommunication Systems Department. He spent a two-year assignment in the IBM Zurich Research Laboratory in Switzerland during 1986-1987. His work at IBM has been in the area of switching architecture and performance modeling of high-performance integrated networks. He is an Editor of IEEE COMMUNICATIONS MAGAZINE, and is currently an Adjunct Lecturer in the Graduate Center at Polytechnic University, NY.

Wolfgang E. Denzel received the M.S. and Ph.D. degrees in electrical engineering from the University of Stuttgart, Germany, in 1979 and 1986, respectively.

Since 1979 he has worked in the field of distributed switching control as a Research Assistant at the Institute of Communications Switching and Data Techniques, University of Stuttgart, Germany. In 1985 he joined the research staff at the IBM Zurich Research Laboratory, Switzerland. From 1985 to 1988 he was involved in architectural design and performance analysis of an integrated circuit- and packet-switching system. Currently, he is responsible for a switching systems group working on broadband switching.