CHAPTER 2

NETWORK PROCESSOR AND SWITCHING FABRIC ARCHITECTURE

2

NETWORK PROCESSOR AND SWITCHING FABRIC ARCHITECTURE

2.1 INTRODUCTION

This chapter describes need of network processor and discusses functions of NPU (Network Processing Unit), followed by basic network processors architecture. It also gives the taxonomy of switching system in general; we review different switching fabric architectures and discuss performance and implementation issues.

2.1.1 Need of Network Processor

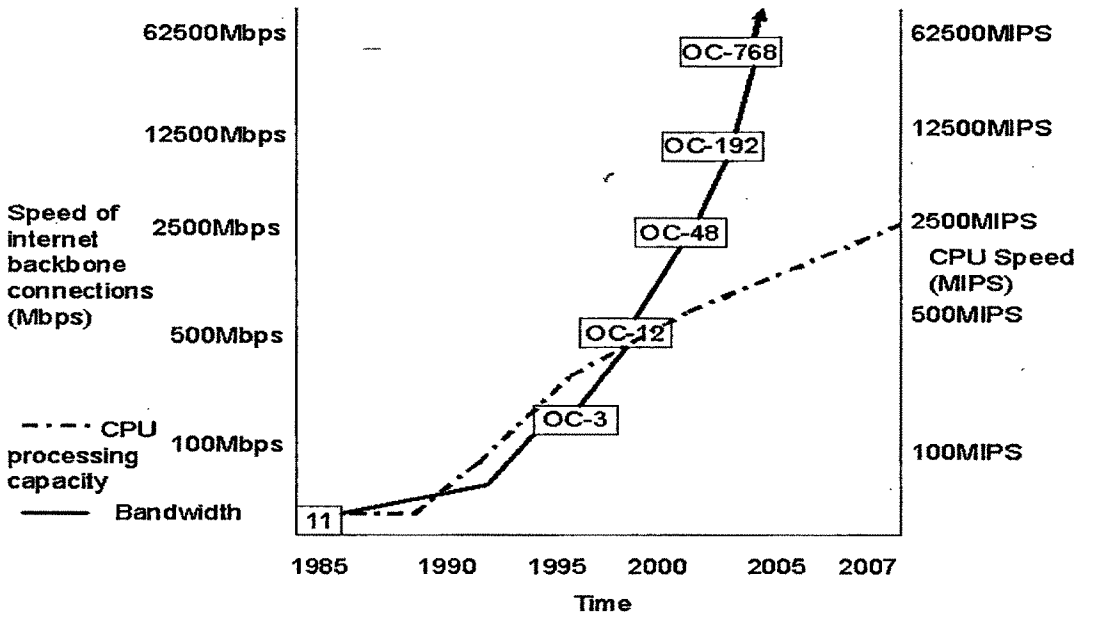


Figure 2.1.Relationship between speed of Internet Connections and time [27]

Network architecture will need enough **flexibility** for tackling new applications like VoIP, MPLS, streaming audio and video, multi-media message services and VPNs. In competitive era continuous innovation in networking and in standards needs **less time to market** and **less time in market.** Network architecture should **support different protocols** and traffic engineering parameters like DiffServe, congestion, security, QoS, address management, etc. To work at line rate it require high throughput.

First generation network system architecture uses General Purpose Processors (GPPs) and third generation network system architecture uses ASIC to implement network functionality. Networking functionality as shown in figure 2.2, can be incorporated into GPP, ASIP, ASIC, Co-processor and FPGA as shown in figures 2.3 to 2.7. The shaded portion indicates functionality implemented whereas, the remaining part indicates the flexibility available within specific architecture.

Figure 2.2.Networking Functionality

1. GPP (General Purpose Processor) – It is a programmable processor for general purpose computing. Slow, expensive general-purpose CPU can provide more flexibility but can not satisfy performance requirement because it is not optimized for the networking application and hence it satisfies networking functionality as shown in figure 2.3.

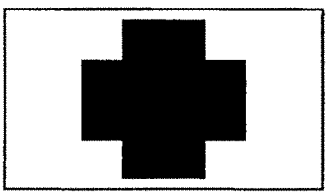


Figure 2.3 General Purpose Processor Architecture for networking functionality

2. ASIP (Application Specific Instruction Processor) – an instruction set processor optimized for a Networking application domain and hence it satisfies networking functionality as shown in figure 2.4, with less flexibility than GPP.

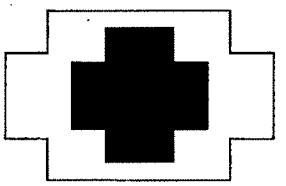


Figure 2.4 ASIP Architecture for networking functionality

3. ASIC (Application Specific Integrated Circuit) –ASICs simply can meet the requirement of performance (due to hardwired solution) but can not meet flexibility requirement because of two long development cycle of ASIC (18 months).The high degree of variability and change at layer 7 makes ASIC implementations less attractive for operations using application-layer information. ASIC satisfies networking functionality as shown in figure 2.5, with no flexibility.

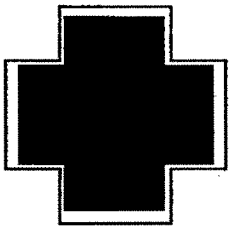


Figure 2.5 ASIC Architecture for networking functionality

4. Co-processor – A hardwired, possibly configurable solution with a limited programming interface normally used to off-load the main processor task. Co-processor satisfies networking functionality as shown in figure 2.6, with moderate flexibility.

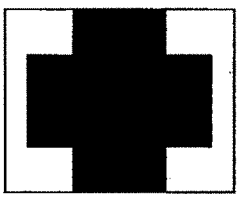


Figure 2.6 Co-processor Architecture for networking functionality

5. FPGA (Field Programmable Gate Array) – A device that can be reprogrammed at the gate level and hence networking functionality can be burnt in it, as shown in figure 2.7, with moderate flexibility.

Unused part of FPGA→→

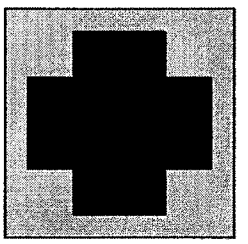


Figure 2.7 FPGA for networking functionality

Figure 2.8 shows the comparison of all the above alternatives in terms of programmability v/s performance.

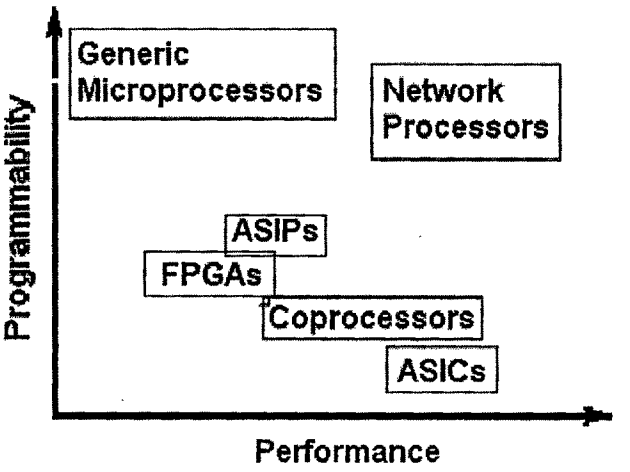


Figure 2.8 Programmability v/s performance aspect

2.1.2 Network Processor Introduction:

In first generation network system architecture, General Purpose Processor (GPP) i.e. CPU is surrounded by Network Interface Cards(NIC). Here hardware architecture used with a software-based network system and most networking functions above the physical layer have been implemented by software running on a general-purpose processor as shown in figure 2.9. The CPU handles all protocol processing tasks except for framing and onboard address recognition.

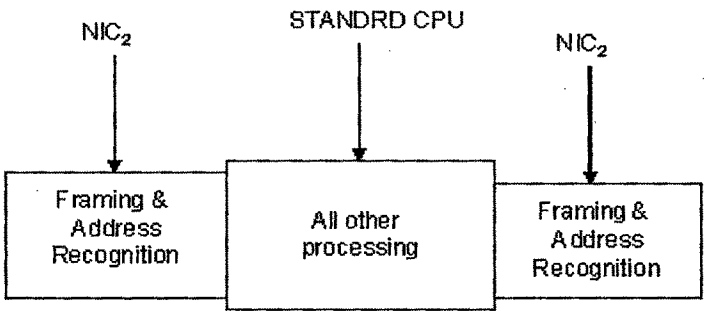


Figure 2.9 First Generation Network System Architecture [15]

In second generation network system architecture, standard CPU manage and control the system, updates routing tables and forward the packets, handles exceptions and errors , including incoming ICMP messages and packets that do not match any classification. Figure 2.10 shows the fast data path between each interface.

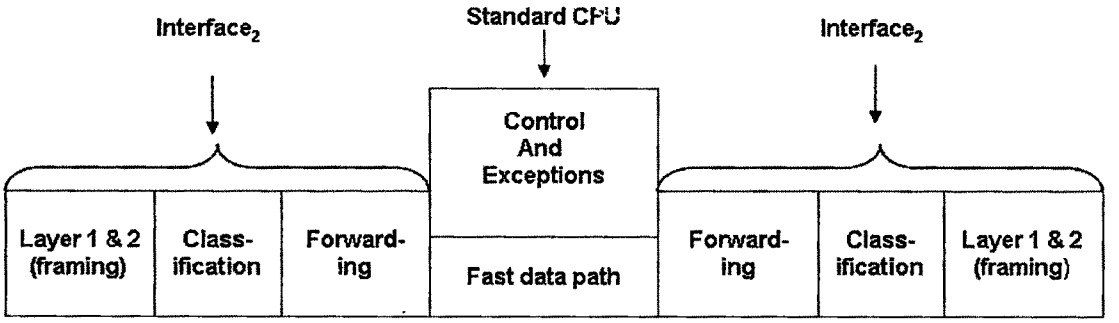


Figure 2.10 Second Generation Network System Architecture [15]

Third generation network system architecture uses specialized hardware to decentralize protocol processing task as shown in figure 2.11. Layers 1 and Layers 2 (physical and framing) are handled with commodity chip sets. ASIC is used to provide basic layer 3 functionality, packet classification and forwarding over the switching fabric. It also provides control and traffic management facilities. A GPP (CPU) handles routing and other tasks that do not lie on the fast data path.

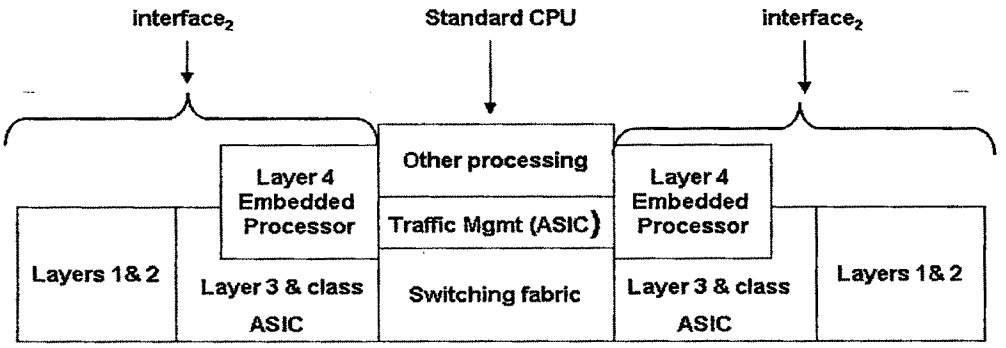


Figure 2.11 Third Generation Network System [15]

With the rapid change in lower layer protocols (e.g. MPLS, DiffServ) and higher layer applications (e.g. Peer-to-Peer, streaming video), this solution will not scale. The need for customizability, in-the-field programmability, and shrinking time to market windows in network processing implementations has created acute need of the network processors.

A network processor is a software programmable device with architectural features and/or special circuitry for packet processing. In broad manner, network processors share characteristics with many different implementation choices:

- network co-processors
- communication processors used for networking applications
- "programmable" state machines for routing

- reconfigurable fabrics (e.g. FPGAs)
- GPPs used for routing

Network Processor Unit (NPU) has ability to correct problems in the field, upgrade features and functions leading to On-System Reprogramability.

2.1.3 Network Processor Functions

Network processing functions could include but are not limited to:

2.1 Classification: Parsing the packet to determine destination and any special processing requirements. Packets can be classified according to the application and priority like voice, multimedia and video. Efficiency of Packet classification is usually evaluated by the Search time, storage space requirement and update time.

2.2 Modification: Changing the contents of the packet, for example, doing encryption for security processing.

2.3 Queuing: Assigning the packet to a queue (specifying priority) for presentation to the switching fabric. Key operations that must be performed in this area include Pointer management, data byte counters, timers (for aging data) and queue servers capable of reordering data from ingress to egress.

2.4 Management and Control: Managing the process, dealing with exceptions, talking to control elements of the switch.

2.5 Content Accessible Memory (CAM): CAM memory implement a map data structure. It is able to compare a given value with all keys in a stored set concurrently.

2.1.4 Basic Architecture of General Network Processor

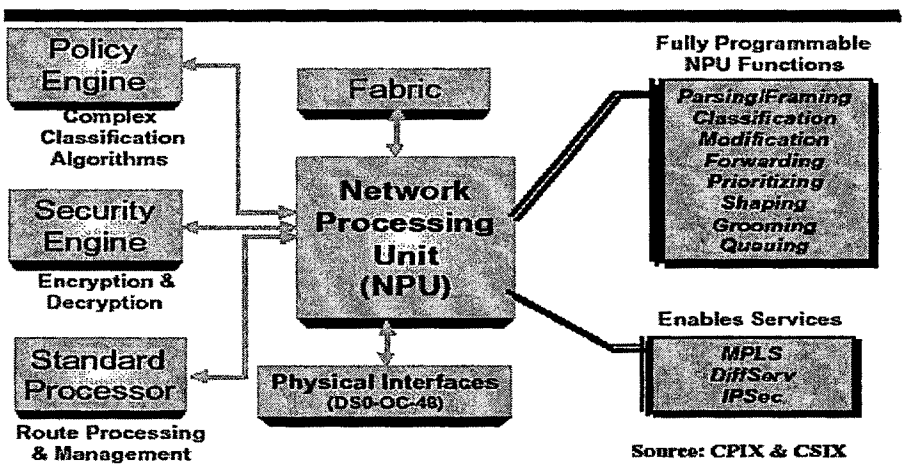


Figure 2.12 Basic Network Processor Architecture [16]

As shown in figure 2.12, in network processor, a core processor manages complex global tasks, while multiple low-level processors (micro engines) perform the packet-processing operations. Network Processors usually consist of multiple processing units such as CPU cores, micro-engines, and dedicated hardware for compute-intensive tasks such as header parsing, table look-up and encryption/decryption. The term *network processor* has been applied to a range of products, from Ethernet-to-DSL controllers to terabit switching. Many favored processors address mid- to high-end switching with speeds that range from OC-3 (155-Mbit/s) enterprise solutions to OC-192 (10-Gbit/s) channels used by ISPs and carriers. This range includes OC-12 (622 Mbits/s), OC-48 (2.4 Gbits/s), and Gigabit Ethernet OC-768, a long-term target for many network-processor vendors.

2.1.4.1 Architectural Issues

1. Distributing the control and data-handling responsibilities across a core processor and multiple micro engines, i.e. (plane of processing data, management and control).
2. Communication Infrastructure that enables the processors to communicate with each other.
3. Throughput [Speed of processing packets (Mbps)] , data rates
4. Trade-offs between parallel engines, context switched multithreading, multiple parallel instructions Very Large Scale Instruction Word (VLIW) architectures.
5. Resource allocation and partitioning.
6. Capability for different packet processing functions and task scheduling options.
7. Hardware Software Co-design strategy.
8. Chip area constraints and System cost
9. Hardware structure of Memory architecture
10. Network processor is a part and parcel of large system so its interface is one of the issues. Open interfaces for ease of networking hardware and software/OS porting (CSIX, PoS Phy, CPIX) is used.
11. Efficiency in instruction and instruction appropriateness for multi-stream packet processing.

Different companies have developed their own proprietary processors; like Agere (PayloadPlus), AMCC Applied Micro Circuits (nP7xxx), EZchip (NP-1), IBM (PowerNP), INTEL(IXP 1200), Motorola C-Port (C5), Vitesse, formerly SiTera (Prism IQ2000) as mentioned in [1] [3] [21] [22] [23] [48] [52] [53] [43].

2.2 REVIEW OF VARIOUS SWITCHING FABRIC ARCHITECTURES

2.2.1 Introduction:

Network with point-to-point links among all the nodes are known as fully connected networks. Fully connected communication networks with N nodes would require L= N (N-1)/2 links. As N increases, there is a tremendous increase in number of links L. Switching allocates transmission facilities like bandwidth and buffer capacity to users to provide them with a certain degree of

connectivity. Recent advancement in VLSI helped in developing fast packet switches for ATM networks. There are various switching techniques, chosen on the basis of optimizing the usage of bandwidth, buffer management, and fabric speed in terms of line speed, number of switching elements, scalability (in terms of data rate, packet rate, and number of ports), throughput, ability to transfer unicast, multicast, and broadcast packets and ease of implementation in VLSI.

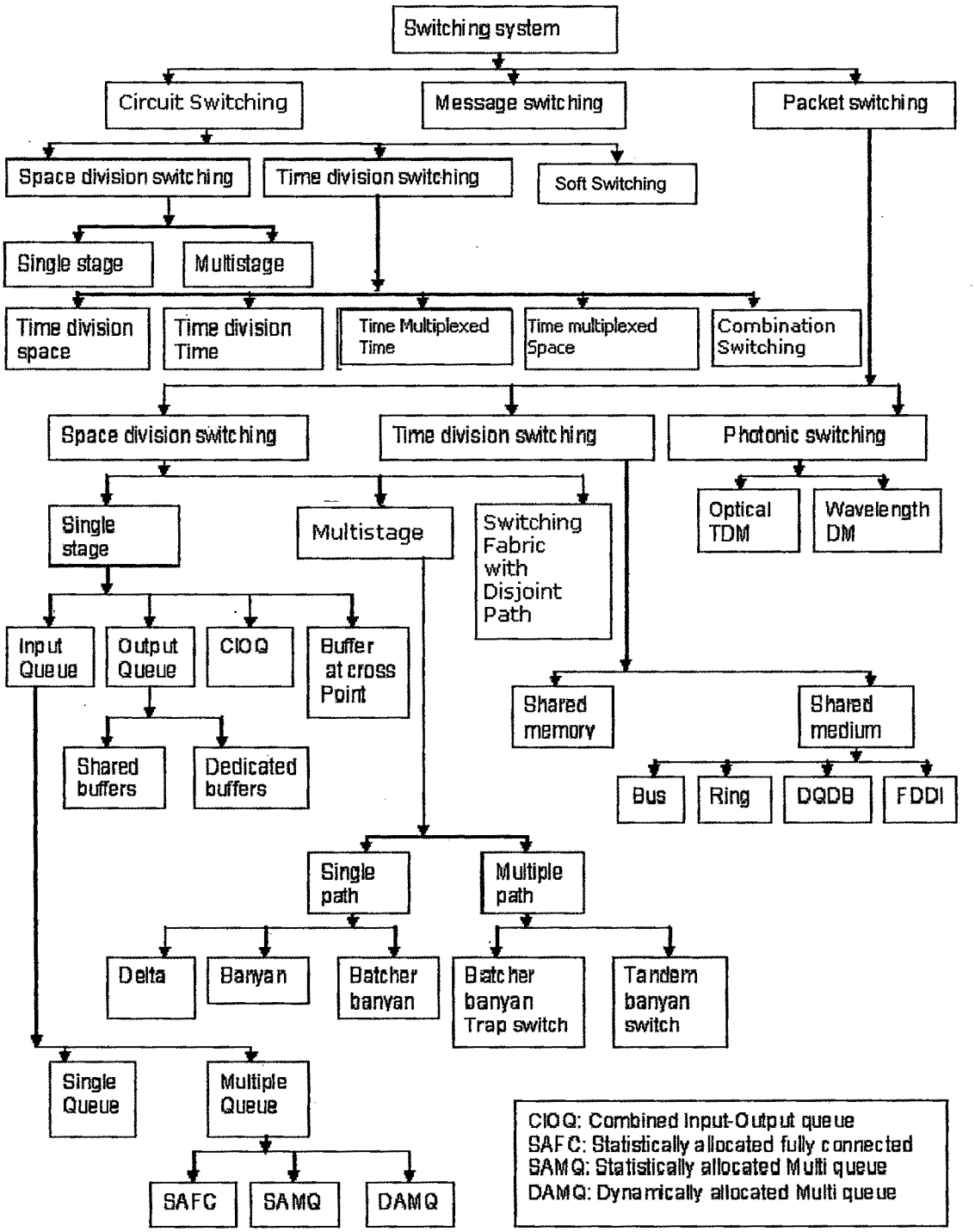


Figure 2.13 Taxonomy of switching system

2.2.2 Taxonomy of Switching System:

Figure 2.13 shows the taxonomy of switching system.

A Circuit switching: A path is setup from source to destination at the connection setup time. Once this path is setup, it remains fully connected for the duration of the connection. Circuit switching was originally designed to handle voice traffic but can handle digital data transmission. In data transmission much of the time the line is idle so bandwidth is wasted.

A.1 Space Division Switching (SDS): Signal paths are divided in space (i.e. physically separate from one another). Each connection requires the establishment of a physical path through the switch, which is dedicated solely to the transfer of signal between two end points.

A.1.1. Single stage SDS: N x N cross bar matrix is as shown in Figure 2.14.

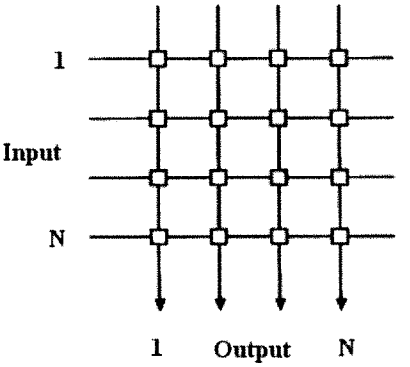


Figure 2.14. Single stage cross points matrix.

Disadvantages:

- The number of cross points grows with the square of the number of nodes.
- If a particular cross point fails, associated connection cannot be established.

A.1.2 Multistage SDS: It is used to reduce the number of cross points, to provide alternative paths and there is no capacitive loading problem.

Disadvantage:

- It is blocking. (Path may not be available to connect an input to output), while single stage SDS is non-blocking. Multi stage SDS can be made non-blocking by increasing the number of intermediate switches. Multiple cross points may degrade the quality of connection.

A.2 Time Division Switching (TDS): Switching element is shared to transmit data from a number of inlets to the corresponding outlets.

A.2.1 Time division space switching: A single switching element, i.e. the bus is being time shared by N connections, all of which can be active simultaneously and a physical connection is established between the inlet and the outlet for the duration of the sample transfer.

A.2.2 Time Division Time switching: If we use a memory block in place of the bus, then the data coming in through the inlets are written into the data memory and later readout to the appropriate outlets. There is no physical connection, even momentarily, between inlets and outlets in the case of data memory based operation. There is a time delay between the acquisition of a sample from an inlets and its delivery to the corresponding outlet.

A.2.3 Time multiplexed space switching: It is used in transit exchanges, where inlets and outlets are trunks, which carry time division multiplexed data streams. There are N incoming trunks and N outgoing trunks, each carrying a time division multiplexed stream of M samples per frame. In one time slot, N samples are switched.

Disadvantage:

- Interchange of samples among different time slots is not possible. So switch does not support full availability. For every input, there are N (M-1) outputs that cannot be reached.

A.2.4 Time multiplexed time switching: Time Slot Interchanging (TSI) is possible in it. So a data sample input during one time slot may be sent to the output during a different time slot. M channels are multiplexed on each trunk. The switch is organized in the sequential write/ random read fashion.

Disadvantage:

- TSI switches are not capable of switching sample values across the trunks without the help of some space switching matrices.

A.2.5 Combination Switching: A combination of the time and space switches leads to configurations that achieve both time slot interchange and sample switching across trunks. A two-stage combination switch may be organized with time switch as the first stage and the space switch as the second stage or vice versa, known as Time Space (TS) and Space Time (ST) respectively. The most common three stage configurations are: Time Space Time (TST), Space Time Space (STS) [99].

A.3 Soft Switching:

In traditional circuit switched networks, hardware and software are not independent. Circuit switched networks rely on dedicated facilities for inter-connection and are designed primarily for voice communications. The more efficient packet based networks use the Internet Protocol (IP) to efficiently route voice and data over diverse routes and shared facilities. Soft switch is a latest trend due to significantly less cost with more functionality. It is the concept of physically separating the network hardware from network software. It is basically a general-purpose computer running specialized software that turns it into a smart phone switch. In soft switch technology, the physical

switching function is performing by a media gate way (MG) and the call processing logic resides in a media gateway controller (MGC).

B. Message Switching: Communication among computers happens in bursts. Message is a block of data with a header that contains control information like source and destination addresses, priority etc. Assigning a continuous connection with high bandwidths for bursty data is waste of resources and results in low utilizations. Also, if the circuit of high bandwidths was set up and released for each message transmission, then the set up time acquired for each message transmission would be high compared to the transmission time of the message.

C Packet Switching: In packet switching, messages are first broken into certain blocks called packets, and then packets are transmitted independently. A packet switch is a box with N inputs and N outputs ($N \times N$) that routes the packets arriving on its inputs to their requested outputs. Switching fabric handles fixed size data units. To switch variable sized packets, segmentation and reassembly functionality should be included. There are three main components (functions) of packet switches: 1) the block that provides the physical connection between the input and output ports (routing), 2) the internal storage, where the packets are stored (buffering), and 3) the scheduling module that determine the departure of packets from the switch (controlling).

Advantages:

- Successive packets in a message can be transmitted simultaneously on different links, reducing the end-to-end transmission delay.
- Due to the smaller size of packets compared to messages, packets are less likely to be rejected at the intermediate nodes due to storage capacity limitation at the switches.
- Both the probability of error and the error recovery time will be lower for packets since they are smaller. Once an error occurs, only the packet with the error needs to be retransmitted rather than the whole message [96].

C.1 Time division Packet switching: In time division Packet switching internal communication paths are shared either by medium or by memory.

C.1.1 Shared Medium Architecture:

C.1.1.1 Bus Architecture: The inputs and outputs of the switch are connected to a single bus or a number of parallel buses to increase the speed, as shown in figure 2.15. Each input and output port assigned a unique address. The inputs have to contend for the control of the bus.

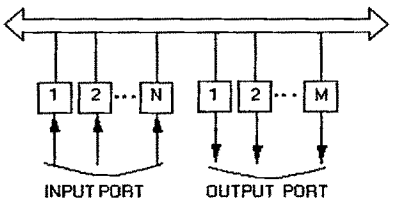


Figure 2.15 A Shared Bus Architecture.

A bus arbitration technique has to be implemented in the bus processor to arbitrate the control of the bus among the input ports.

Disadvantages:

- If the input/output line rate is R and there are n ports, then the bus should have a minimum speed of Rn. So, for a bus clock of r Hz, the bus has to be $w = Rn/r$ bits wide. (i.e. bus speed has to grow with the number of links).
- Problem of capacitive loading on the signal lines rises as the number of ports connected to the bus increases. This reduces the maximum clock frequency of the bus.

C.1.1.2 Ring Architecture

Ring architecture replaces the bus with a ring-based interconnects.

Advantage:

- Ring systems, have quadratic growth characteristics, but unlike buses, they are not further limited by capacitive loading and other electrical transmission effects. Because all data transmission is point-to-point, one can typically operate rings with substantially higher clock frequencies than bus systems implemented using the same technology.

Disadvantage:

- Rings do introduce some additional latency, relative to buses, but for switching applications these latencies are typically fairly modest.
- Determining the bandwidth needed to support fully non-blocking operation is more complicated for a ring than for a bus.

C.1.1.3 DQDB: Distributed Queue Dual Bus (DQDB) is an IEEE standard for communication in Metropolitan Area Networks (MAN). That uses a pair of unidirectional buses and a multiplexing scheme that assigns a time slot to each sender. It has cell granularity, which means that a large packet must be divided into cells for transport.

C.1.1.4 FDDI: Some fabrics have been built using Fiber Distributed Data Interconnect (FDDI) as a shared medium. FDDI uses a ring topology and token passing to control access.

C.1.2 Shared Memory Architecture (SMA): When SMA has a packet to send, an input port deposits the packet in memory and controller informs the appropriate output port that packet is ready. The message specifies the address in memory where the packet resides. The output port extracts the packet from memory. The memory bandwidth, access time and size are important factors in this switch design. If we define S as the port speed, say in bits per second (bps), then the memory bandwidth should be at least 2NS. Each port needs memory interface hardware that connects the port to the memory system, which is expensive. To implement switch at low cost, multiple I/O ports can share a single memory interface. A separate memory interface for each port is

also possible to get maximum performance.

Disadvantage:

- The shared memory must operate N times faster than the port speed because cells must be read and written one at a time. As the access time of memory is physically limited, the approach is not very scalable.

C.2 Space Division Packet Switching

C.2.1 Single Stage (Crossbar architecture) for Packet Switching: It is a matrix-like space division fabric in which the interconnection between an input port and an output port is switched by switching fabric consists of an N x M array of switches. It includes controller hardware that handles port contention by ensuring that only one input port accesses each output port at any time. Buffering (queuing) in packet switch is necessary because packets that arrive at interconnect are unscheduled and the switch has to multiplex them and sometimes packets compete for the same output port.

Advantages:

- Crossbar switches provide simultaneous connections between input and output ports, allowing multiple cells to pass from Input ports, through the crossbar to output ports at the same time. A controller can establish up to P simultaneous paths, where P= min (N, M).
- Input and output ports can operate at lower bandwidths than required in systems using buses or rings.
- A crossbar with N inputs and M outputs contains a total of N x M (N^2 if N=M) cross points. Thus, the crossbar has quadratic scaling characteristics, like bus and ring systems. However, with a crossbar, the part that scales quadratically is the number of cross points, while in buses and rings, it is the number of pins and associated interface circuits in the input and output ports. Since cross points are very simple circuit components and consume a small amount of chip area, the quadratic growth characteristic is less limiting for crossbars than it is for buses and rings [98].
- Crossbars are non-blocking, which means any input-output pair can talk to each other as long as they do not interfere with other input-output pairs.

Disadvantages:

- Only single cross-point for particular input/output pair, no other alternative path.
- Crossbars are expensive because it requires NXM cross points.

C.2.1.1 Output Queued (Buffer) (OQ): An output buffer, which is located between the fabric and the output port, allows bursts from two or more input ports to arrive at a single output port (i.e. receive multiple cells per cell time) as shown in figure 2.16. Here, either the output buffers must operate at some factor times the port speed, or there should be multiple buffers at each output. In

both cases, the throughput and scalability are limited, either by the speedup factor or by the number of buffers. Decision of optimum queue size is necessary in output buffering case. A small queue can not absorb a burst of packets, so packets will be dropped. A large queue will introduce excessive delay such that packets are no longer valid.

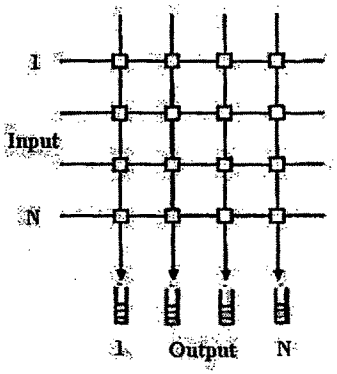


Figure 2.16 Output buffer

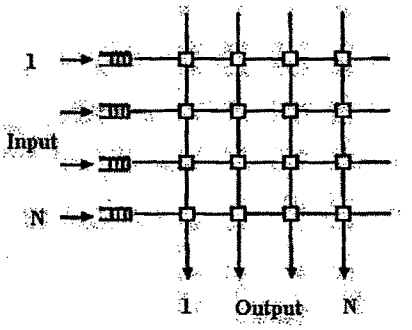


Figure 2.17 Input buffer

We can share the output buffers by two approaches.

1. Complete Partitioning (Dedicated buffers): Equally partitioning the output memory between the output lines.
2. Full Sharing: This approach pools all output memory into one completely shared buffer rather than have a separate buffer for each output, known as full sharing.

C.2.1.2 Input Queued (Buffer) (IQ): If a burst of packets arrives at an input port, the port must send each packet quickly or packets will be lost. An input queue can be placed between each input port and the fabric, which stores succeeding packets, while the packet at the head of the queue is, blocked waiting for access to an output port as shown in figure 2.17.

C.2.1.2.1 Single Queue IQ:

Advantage:

- Buffer needs only a single write port to receive single packet at a time
- The fabric and memory of an IQ switch need to be merely as fast as the line rate. So suitable for switches with fast line rates or with large numbers of ports. It requires smaller system bandwidths.
- For multicast traffic (traffic that is sent from a single input port to multiple output ports), a burst of n cells that are to be delivered to m output ports only needs n cell buffers for the IQ structure, rather than $m \times n$ buffers for OQ structure.

Disadvantage:

- Head of line (HOL) blocking is a major disadvantage of single queue IQ switches. It occurs when a packet at the head of queue, waiting for a busy output, blocks a packet behind it that is

destined to an idle output. It can have the worst effect when the traffic is periodic and the scheduling algorithm is based on priority rotation [84].

C.2.1.2.2 Multiple Queues IQ:

For IQ switches to be efficient, we must overcome the limitations of HOL blocking by using Virtual Output Queuing (VOQ) as shown in fig 2.18. In this scheme each input has *N* queues or blocks of memory instead of one single FIFO queue.

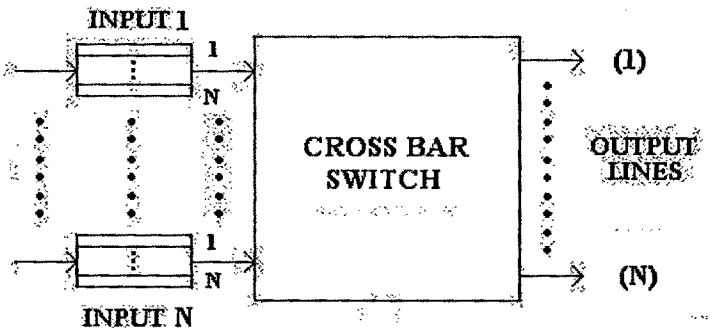


Figure 2.18 Virtual Output Queuing structure.

I Statically allocated, fully connected (SAFC) buffer: Packets must be segregated according to the output port to which they have been routed. Separate FIFO buffers are used for each of the output ports at each of the input ports, as shown in Figure 2.19.b. Here packets in every queue are contending for the same output. Hence, the packet at the head of line is not blocking a packet behind it from being sent to an idle output (and hence no HOL blocking exists). Every input can send *N* packets in every time slot (rather than one packet in case of single FIFO inputs). This increases the throughput of the switch.

Disadvantages:

- Four separate switches must be controlled, as opposed to a single-crossbar.
- Each input port will require *N* separate buffers and buffer controller.
- Buffer utilization is inefficient compare to FIFO switch. As the available buffer space at each input port is *statically* partitioned so that, for 4×4 switch, only one quarter of the input buffer space is available as potential storage for any given packet.
- Pre-routing is required for every packet in order to determine the destination output port (and hence the input queue the packet belongs to) [67].

II Statically allocated multi -queue (SAMQ) buffer: The SAFC buffer can be simplified by implementing the four separate buffers at each input port as a single buffer whose space is divide into four separate queues (Figure 2.19.c). This does not reduce the rate at which the buffers can receive packets; since there is only single input port supplying all *N* queues.

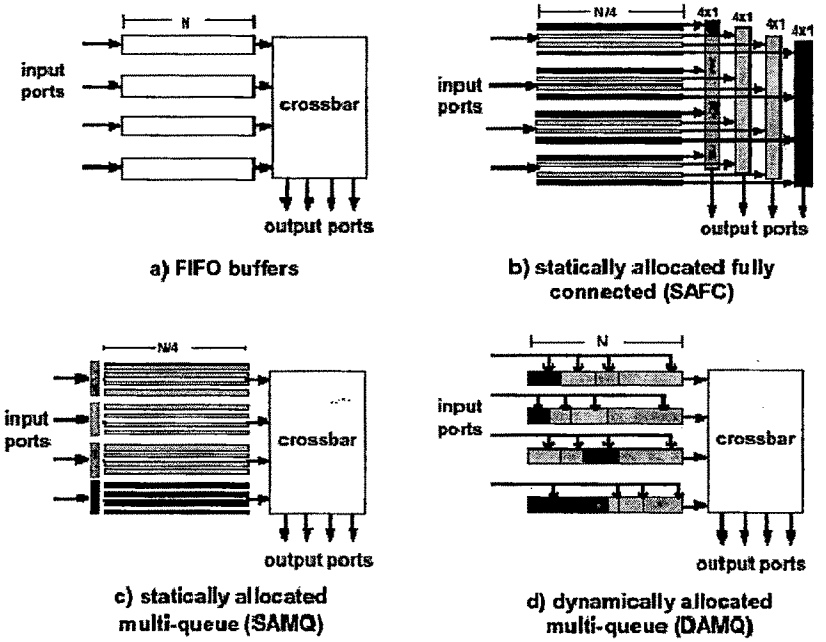
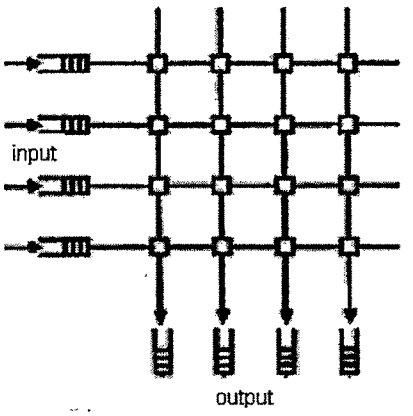


Figure 2.19: Switches with multiple input queues [61]

This eliminates some of the overhead associated with the SAFC switch, and reduces the number of packets, which can be read from the queues associated with an input port (assuming that the buffer has a single read port and a single write port). However, the problem caused by the need to preroute packets and the inefficiency of statically partitioning the available buffer storage are the same as in the SAFC switch [61].

III Dynamically allocated multi -queue (DAMQ) buffer: As shown in Figure 2.19.d, it has none of the disadvantages mentioned earlier. Each input buffer uses a single buffer pool. Virtual queues are allocated dynamically within each input buffer and that makes the buffer usage more efficient. Disadvantage: Controlling is complex.

C.2.1.3 Combined input/output queuing (CIOQ): When we operate the switch fabric at a faster speed than the input/output lines, it reduces the effect of HOL blocking but does not remove it completely [12]. A speedup by a factor of *S* can remove *S* packets from each input port within each time slot. The switches that use speedup S >1 may lead to accumulation of cells at the output ports so both input and output buffers are required as shown in figure 2.20. CIOQ switch uses backpressure as the means to inform the fabric arbitration algorithm about the state of the output queues.



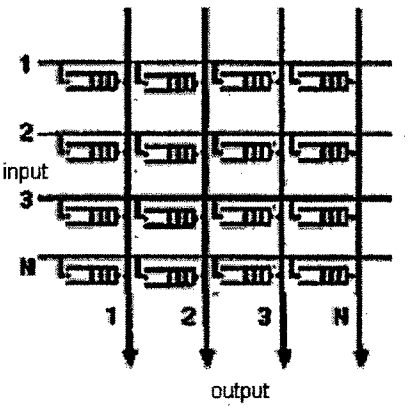


Figure 2.20 Combine input output buffer **Figure 2.21 cross point buffer**

C.2.1.4 Buffers at Cross-point: Placing the buffers at the individual cross points as shown in figure 2.21, does not suffer the throughput limitation incurred with either dropping packets or input buffering. It is similar to achieving output queuing, but queue for each output is distributed over *N* buffers.

Advantage:

- Each buffer must have a less bandwidth compared to the shared-bus case.

Disadvantage:

- Buffer memory typically requires a much larger real-estate than the switching array itself, and combining them both on the same circuit would limit the size of the switching fabric.
- The total memory required for a given loss rate is greater than that required for output queuing with complete partitioning, because buffers are not shared.

C.2.2 Multistage architecture Multistage Interconnection Network (MIN) architecture lies at intermediate points between the extremes of a single shared path (e.g. a shared medium) and separate paths for each pair of ports. (e.g. a cross bar). In general, a multistage network consists of n stages to connect N input lines, where $N=2^n$. Each stage may use N/2 switch elements. So an $N \times N$ Banyan switch uses $(N/2) \log_2 N$ elements. Multistage switches can be either Blocking, Rearranging non-blocking and non-blocking. There are two types of blocking: Internal Blocking and Output Blocking. Packets destined for different outputs may compete for a particular link inside the network is known as ***Internal Blocking***. Packets compete for the same output port, is known as ***Output Blocking***. Network can perform all possible connections between inputs and outputs by rearranging its existing connections, is known as ***Rearranging non-blocking***. Network can handle all possible connections without blocking is known as ***Non-blocking***. Basic switch element is essentially an interchange device with two inputs and two outputs. Figure 2.22 shows two functions of a switch element: Bar (straight) and cross (exchange). If both inputs require the same output line,

then only one of them will be connected and the other will be blocked or rejected or in some cases it can even be buffered. This capability of routing the packet through the switch fabric based on the address field contained in the packet header, without depending on a central controller, is called self-routing.

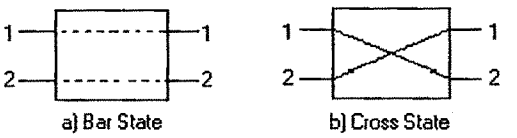


Figure 2.22. Basic switching element

C.2.2.1 Single path

In single path networks, there is only one path to the destination from a given input. Routing is simple because there is only one path exists to the proper output.

(i) Delta Network: Delta Network is defined as an a^n x b^n switching network with n stages, consisting of a x b switching elements, as shown in figure 2.23. The only rule to follow during construction of the network is that if a switching element has its inputs coming from other switching elements, then both inputs must come from either upper lines of preceding-stage or both must come from lower lines of preceding-stage switching element. No input or output line of any switching element is left unconnected. There are numerous types of delta networks, such as rectangular delta networks (where the switching elements have the same number of outputs as inputs), omega, flip, cube, shuffle-exchange (based on a perfect shuffle permutation) and baseline networks.

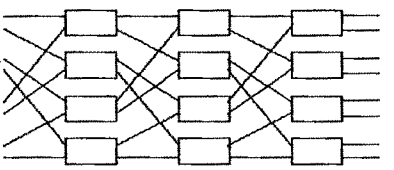


Figure 2.23 8×8 Delta network (Omega)

(ii) Banyan Network The Banyan network is constructed of an interconnection of stages of switching elements. A basic 2x2 switching element can route an incoming cell according to a control bit (output address). If the control bit is 0, the cell is routed to the upper port address; otherwise it is routed to the lower port address as shown in figure 2.24. Banyan switch is modular, (Scalable) i.e. a large Banyan switch is created by interconnecting smaller Banyan switches. Interconnections between stages follow the concept of deck shuffling of playing cards [84].

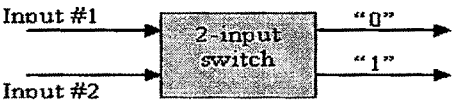
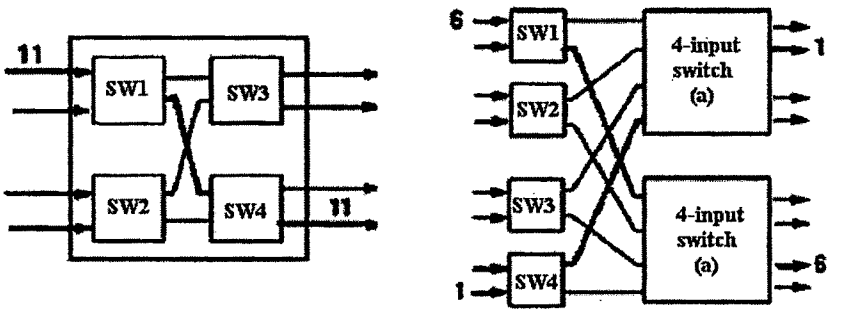


Figure 2.24 2-input Banyan switching element.

The operation for input control data 11 is shown in figure 2.25(a). The structure of 8-input Banyan

switch and the operation for input control data 6 and 1 is shown in figure 2.25 (b).



(a) (b)

Figure 2.25 (a) 4 x 4 banyan (b) 8 x 8 banyan

(iii) Batcher-Banyan architecture: When internal blocking occurs in Banyan Network, only one of the two cells contending for a link can be passed to the next stage, so overall throughput is reduced. This problem can be solved by sorting the cells according to the output they are destined for, before sending them into the banyan. Such architecture, called ***Batcher-Banyan architecture***, has been used in the Sunshine switch [15]. Incoming cells are sort, based on the Bitonic sort algorithm. A Batcher switch is built up of 2 x 2 switching elements, but these work differently than those in Banyan switch. When a switching element receives two cells, it compares their output address numerically (thus not just 1 bit) and routes the higher one on the port in the direction of the arrow, and the lower one the other way. If there is only one cell, it goes to the port opposite the way the arrow is pointing.

Disadvantage:

- Batcher-Banyan cannot handle output blocking.

C.2.2.2 Multiple path

In multiple path networks number of alternative paths exist to the destination output from a given input, so internal blocking can be reduced or avoided. In multiple path networks the internal path will be determined during the connection set up phase and all cells on the connection will use the same internal path.

(a) Batcher Banyan trap switch (Starlite Switch):

In case of output blocking, the only solution to the problem is buffering. We can use a recirculating buffer external to the switch fabric. Here output conflicts are detected after the Batcher sorter, and a trap network selects a cell to go through and recirculates the others back to the inputs of the Batcher network. The purpose of the concentrator network as shown in figure 2.26 is to reduce the number of input lines to the sort network. This is possible because a significant number of users will be idle.

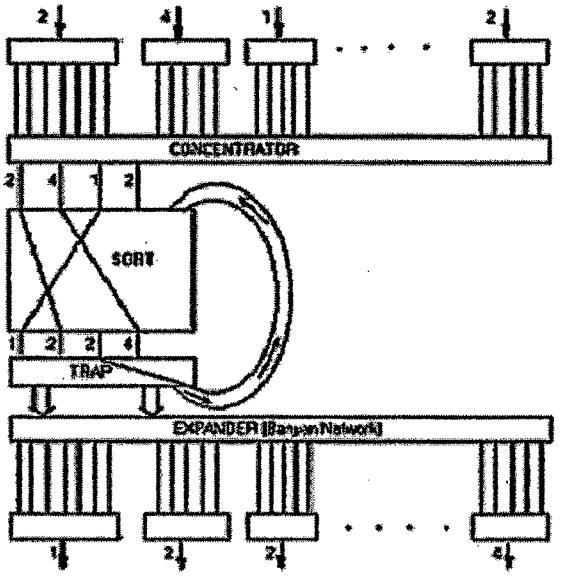


Figure 2.26 Starlite switch [84]

Disadvantage:

- This approach requires complicated priority control to maintain the sequential order of cells and increases the size of the Batcher network to accommodate the recirculating cells [66].

(b) The Tandem Banyan Switch:

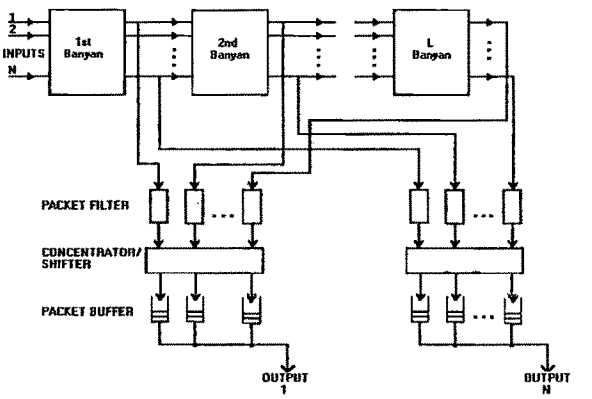


Figure 2.27 Tandem Banyan Switch [49]

Only difference between the Tandem Banyan switch shown in Figure 2.27 and the Knockout switch is the replacement of the N input broadcast lines by a cascade of memory less L Banyans. Consequently, there are only L packet filters per each output. The jobs of the concentrator, the shifter, and the shared buffers remain the same as for a knockout switch. The first Banyan does the best it can to route its inputs to its correct outputs. However, since the path from every input to any output inside the Banyan is unique, contention may arise if two inputs either request the same output or need to use the same internal link. In both cases, one of the cells is routed correctly, while

the other one is misrouted and appears on a wrong line at the output. In order to minimize the damage caused by misrouted cells, any cell that has been misrouted once is marked so as not to contend for internal links in later stages of the Banyan, which can be needed for correctly routed cells. Every cell that has reached its correct output by the end of the first Banyan is intercepted by the first packet filter of that output module. Only the cells misrouted by the first Banyan enter the second one, and so on; each subsequent Banyan receives only cells misrouted by previous Banyans in the cascade. Cells still misrouted after L stages are irrecoverably lost [49].

Disadvantage: It cannot support multicasting as easily as the knockout switch does. It does not become independent for large N, as was the case in knockout switch.

C.2.3 Switching Fabric with disjoint path (the Knockout Switch): The knockout switch as shown in Figure 2.28 consists of packet filter, concentrator, shifter, and output queues. The job of each packet filter is simply to pass the cell to the concentrator if the cell is destined for that output, and to mark the cell as inactive otherwise. The role of the concentrator is to identify among its inputs those cells that are active and route them to its leftmost outputs, one cell per output line. It is not really necessary to buffer up to N cells at each output. Because it is generally highly unlikely that all N inputs will have a cell destined for a given output within each cell interval.

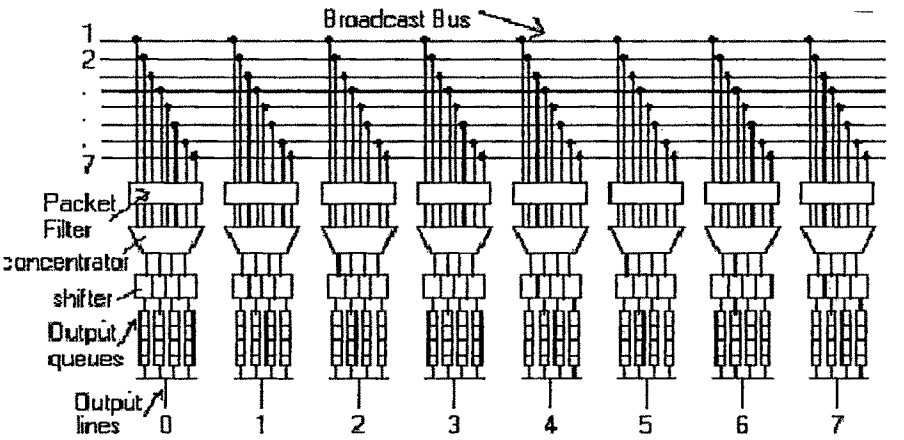


Figure 2.28 Knockout switch architecture [89]

The number of cells accepted at each output port in each time slot can be limited to a relatively small fixed number L, where $L<N$. Should $L+1$ or more cells arrive simultaneously, only L of them will be processed via the concentrator; all others will be lost. The leftmost buffers would tend to fill up faster, and might even overflow despite the presence of empty buffer entries on the right. The shifter prevents that from happening by spreading each bulk of cells arriving at its input continuously to the right; Because of the round-robin nature of the shifter and the fact that the buffers are filled cyclically, they can also be emptied cyclically. Multicast and broadcast cells are readily supported by knockout switch.

C.3 Photonic Switching: The switching system that interconnects a large collection of fiber optic cables can be divided into three parts: 1) Optical to Electrical conversion (O/E), 2) Electronic Fast Packet Switch and 3) Electrical to Optical Conversion (E/O). Photonic switches avoid the need for O/E and E/O conversion by switching optical signal directly, but control of the network is electronically implemented. Photonic switching uses photonic devices rather than electronic devices to make or break connections within integrated circuits.

2.3 INTEGRATION OF SWITCHING FABRIC WITH NETWORK PROCESSOR.

Switching fabric provides interconnection among input/output ports and scales to handle a high data rate on given port. Different vendors have designed and manufactured a different products for switching fabrics like Hypertransport, Infiniband, PCI-X, Packet Over Sonet (POS), Rapid IO and utopia. Each architecture represents a trade-off among performance (i.e throughput, average latency, delay variance), scalability and cost.

Network processor is a hardware device that can augment or replace any of the functional units in a network system. So NP can perform any of the following architectural roles.

1. Replacement for a conventional CPU

NP must have normal capability plus special instructions and architecture for packet processing. NP requires to access packets from memory, parse packet headers, and used linked data structure with pointers.

2. Augmentation with conventional CPU

In this case CPU performs; most packet processing and network processor handles specific tasks. NP can be augmented either as a pre-processor or as a co-processor .As a preprocessor NP act as a inline filter to handle incoming packets before they arrive at the CPU. So NP retrieves a packet from a hardware port and performs ingress operations such as classification.

In co-processor method, CPU can choose to use a NP for any operation that the NP can perform like packet encryption.

3. On the input path of a NIC

NP on the input path, must able to retrieve packets from an input port and perform ingress operations.

4. Between NIC and switching fabric

The crossbar switching fabric provides switched paths, so path must be established before the fabric can be used. So NP interacts with the fabric controller to establish a path, transfer data and relinquish the path.

5. Between the switching fabric and an output interface

When switching fabric uses a distributed control mechanism instead of central mechanism, NP is

suitable between the switching fabric and output interface. Here each output port controls its own access, when an input port is ready to use the fabric, the input port notifies the output port, by using separate mechanism like independent bus. The output port, schedules request and notifies each potential sender when the fabric is ready for the transfer. NP can co-ordinates access of the output port.

6. On the output path of a NIC

NP on the output path performs egress operations. NP must be able to accept packets from the switching fabric, manage queues, implement traffic shaping, and send packets to the output port.

7. Attached to the switching fabric like other ports

Packets can be sent from an input port across the fabric to the NP and from the NP across the fabric to an output port. Because fabric allows scaling, the set of intermediate NP can be expanded easily in this approach. To permit maximum parallelism, the system can arrange to distribute incoming packets among the entire set of network processors [15].

2.4 SUMMARY

We have reviewed first generation to fourth generation (i.e. Network Processor) network system architectures in first part. In second part, we have discussed taxonomy of switching system. Switching architectures are reviewed and discussed in terms of various aspects like: bandwidth, buffer management, and fabric speed in terms of line speed, number of switching elements, scalability, throughput, and ease of implementation in VLSI. Multicasting is easy to implement in output or shared buffer switch, but output or shared buffer switch requires switch fabric speed N times the line rate, hence it is not suited for fast packet switches. In input queuing, switch fabric speed is same as line rate but maximum throughput of FIFO input buffer is limited to about 58% due to HOL blocking. [56] Throughput can be increased by providing a separate queue for each output port, so total N^2 queues are required. Number of queues can be reduced by dynamically sharing the queue instead of a separate queue. Various scheduling algorithms are available for VOQ. Memory speed is major bottleneck and in input queuing at most only one memory read and one memory write operation is done per port in a single switch slot, which simplifies VLSI design. In multistage architecture numbers of switching elements are reduced from N^2 to $N\log_2 N$, but this is not big advantage due to VLSI technology. A multistage architecture also introduces blocking. Photonic (optical) switches will be the most promising fast packet, because it saves conversion time from O/E and E/O. In the third part, we have discussed various alternatives to integrate switching fabric with NPU.