CHAPTER 15

OPTICAL PACKET SWITCHES

Introduction of optical fibers to communication networks has caused a tremendous increase in the speed of data transmitted. The virtually unlimited bandwidth of optical fibers comes from the carrier frequency of nearly 200 THz [1]. Optical networking technology, such as add-drop multiplexers [2, 3], reconfigurable photonic switches [4], and wavelength multiplexing division (WDM), has progressed well and facilitated optical networking in the past few years [5, 6]. Especially, recent advances in dense wavelength multiplexing division (DWDM) technology have provided tremendous bandwidth in optical fiber communications [7]. However, the capability of switching and routing packets at this high bandwidth (e.g., 1 terabit/s) has lagged far behind the transmission capability. Building a large-capacity packet switching system using only electronic technology may potentially lead to a system bottleneck when interconnecting many electronic devices or modules, mainly caused by the enormous interconnection wires and the electromagnetic interference they would generate. With the advancement of optical devices technology, several packet switch architectures based on WDM technology have been proposed for large-capacity packet switches. Although today's optical packet switching technology is still very primitive and cannot compete with electronic switching technology, optical packet switches have great potential to scale-up their switching capacity as the technology of some key optical devices matures.

A photonic packet switch may require optical devices such as lasers, filters, couplers, memories, multiplexers, demultiplexers, and so on. At the present time, some optical devices are either very power-hungry or too slow in switching compared with electronic devices. However, it is possible to design high-capacity switches by the use of both electronic and optical technologies. In such switches, data transfer can be achieved through optical medium, and complicated functions such as contention resolution and routing control can be performed electronically. These switches are called hybrid switches. The hybrid

High Performance Switches and Routers, by H. Jonathan Chao and Bin Liu

switches that only convert a packet cell header into electronics for processing and controlling but leave the entire cell to be handled in the optical domain are called optically transparent.

The ongoing research into photonic packet switches is to develop faster and larger optical switches and new techniques that can be used to enhance the existing optical switch architectures. There are many issues to be considered when designing an optical packet switch, such as characteristics of the optical devices employed, scalability of the switch, power budget of the system, synchronization between electrical and incoming optical signals, performance of the switch under various traffic patterns, and so on. In addition, some of the techniques developed for optical packet switches could be applied to large scale packet switches where small electronic switch modules are interconnected by an optical interconnection network.

The techniques of space-division multiplexing (SDM), time-division multiplexing (TDM), and WDM have been used in designing optical switches. SDM requires a large number of binary switching elements. From the switch size and cost point of view, it is not an ideal approach for photonic switching. TDM is a more classical technique used in communications [8]. When it is applied to optical switching, complicated temporal compression and temporal expansion circuits are required. The throughput of such a switch is limited by the speed of the demultiplexer, which is actually controlled by electronics for the time being. WDM is made possible by the range of wavelengths on an optical fiber. WDM splits the optical bandwidth of a link into fixed, nonoverlapping spectral bands. Each band has a wavelength channel that can be used for a specific bit rate and transmission technique, independent of the choices for other channels.

In this chapter, we review several approaches to build a large-capacity packet switch and discuss their advantages and disadvantages. Depending on whether the contended packets are stored in the optical or in the electrical domain, these switch architectures are classified into opto-electronic packet switches (described in Section 15.1) and all optical packet switches (described in Section 15.4). Two opto-electronic packet switches are described in detail in Sections 15.2 and 15.3 to better understand switching operations and implementation complexity. Sections 15.5 and 15.6 describe optical packet switches using shared fiber delay lines for optical memory in single-stage and three-stage cases, respectively. In all the architectures presented here, switch control is achieved electronically since, for the time being, it is still complicated to realize logical operations optically. The capacity of electronic control units and the tuning speed of optical devices are the main performance-limiting factors in these architectures.

15.1 OPTO-ELECTRONIC PACKET SWITCHES

For the opto-electronic packet switches, optical switching networks are used for interconnection and transmission between electronic input and output modules. Logical control, contention resolution, and packet storage are handled electronically.

15.1.1 Hypass

HYPASS [9] in Figure 15.1 is an opto-electronic hybrid cell switch in which electronic components are used for memory and logic functions and optical components are used for routing and transporting data. In this figure, bold continuous lines represent optical paths, bold dashed lines represent serial data paths, dotted lines are tuning current paths, and thin continuous lines are control signal paths. The switch is composed of two networks: the

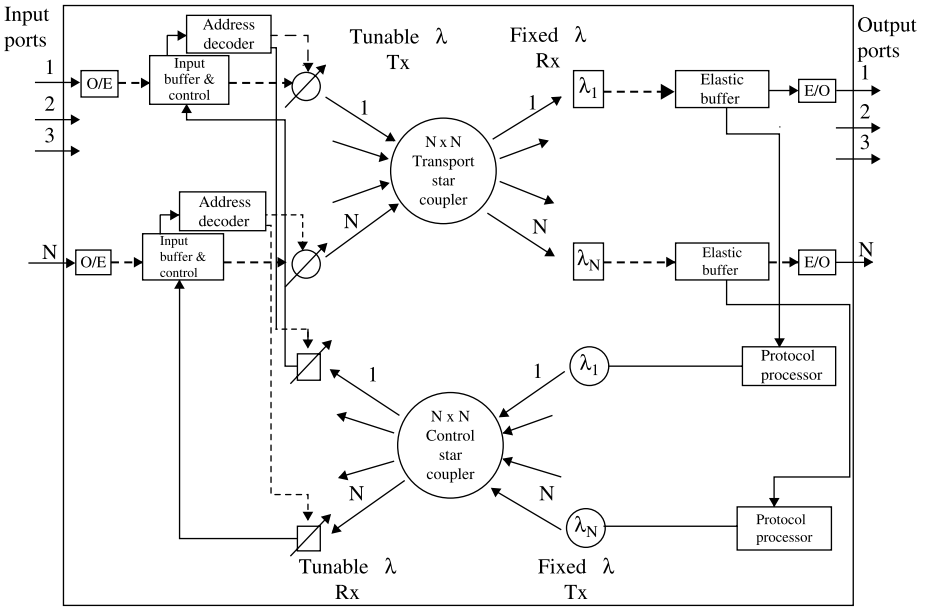


Figure 15.1 Global diagram of the HYPASS implementation (© 1988 IEEE).

transport network and the control network. The architecture is based on the "broadcast and select" approach in both of the networks. There is a unique optical wavelength associated with each of the output ports. As shown in Figure 15.1, the transport network has tunable-wavelength laser transmitters at the input side, fixed-wavelength receivers at the output side, and an $N \times N$ star coupler that transfers the incoming data from inputs to outputs. In order to transfer control information from output ports to the input ports, a similar network is used.

When a cell arrives at an input port of the switch, it is first converted from optical to electronic and its destination address is obtained. Then, the cell is temporarily stored in the corresponding input buffer. The tunable wavelength laser transmitter of the corresponding input port is tuned to the wavelength of the requested output port. When a request-to-send signal (or poll) is received from the corresponding output port via the control network, the cell is transmitted through the transport network. The acknowledgment representing successful delivery of the cell is also transmitted through the control network. If there are multiple cells for the same output port, contention occurs. Power threshold detection or multiple bit detection on the cell preamble could be used to detect collision. The cells that do not get acknowledgments in a slot time are kept to retry later. In order to resolve contention and provide successful transmission of cells, the tree-polling algorithm (explained in [9]) is employed in the selection of inputs in the following cell slots. The cells that reach the output ports successfully are stored in the elastic buffers and transmitted over the optical fiber trunks after the necessary electrical to optical conversion.

The HYPASS architecture has advantages due to its parallel structure. However, since a slot time is based on the length of the polling step, transmission of a cell, and receipt of the acknowledgment, the time overhead for the electronic control and optical tuning operations

are the factors that limit capacity. The switch does not have a multicasting capability due to the usage of fixed wavelength receivers at the output ports.

15.1.2 Star-Track

Star-Track [10] is another hybrid switch architecture. It is based on a two-phase contention resolution algorithm. It also supports multicasting. As shown in Figure 15.2, the switch is composed of two internal networks: an optical star transport network and an electronic control track surrounding the star network. The optical transport network has fixed wavelength optical transmitters at the input-ports side, and wavelength tunable optical receivers at the output-ports side. There is a unique wavelength associated with each input port. Input and output ports are connected through an optical star coupler. The output port conflicts are resolved by the ring reservation technique (see Section 8.3.2). The electronic control network that implements the ring-reservation technique is the major track linking input ports, output ports, and a token generator, sequentially.

Cells arriving at the input ports are stored in the input buffers after optical to electronic conversion. There are two control phases in a cell transmission cycle. In the first phase, input ports *write* their output port requests into the tokens circulating in the control network. The output ports *read* the tokens and tune their receivers to the appropriate input port wavelengths in the second phase. Then, the cell transmission starts over the star transport network. The transmission and control cycles are overlapped in time in order to increase throughput. Since each input has a unique wavelength and there is input–output port pair scheduling prior to transmission, cells are transmitted simultaneously without causing contention.

This architecture allows multicasting. However, the throughput of the switch may degrade as the number of multicasting connections increases due to output port collisions in the first phase. It is shown that this problem can be alleviated by call splitting (i.e., allowing a multicast call to be completed in multiple cell slots). This architecture can support different priority levels for the cell by adding minor tracks into the control network. However, in this case, the token should recirculate among the input ports more than once depending on

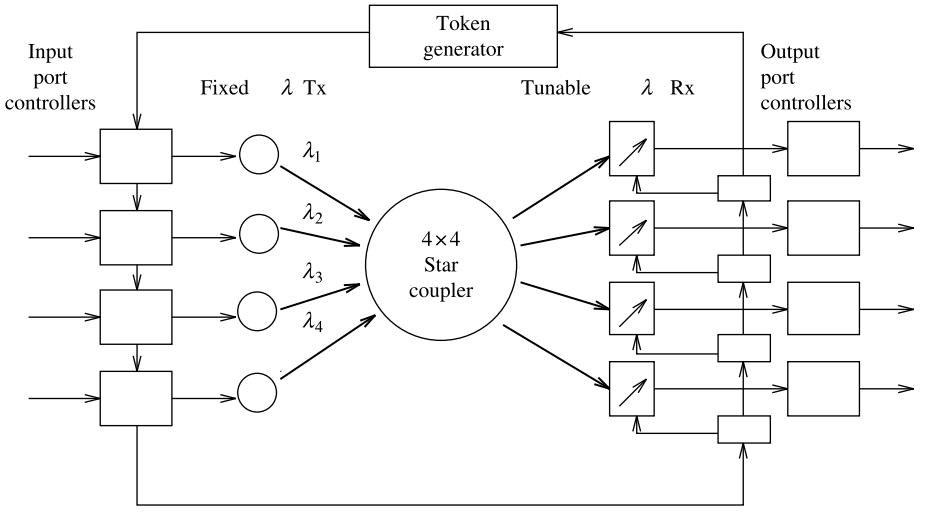


Figure 15.2 Star-track architecture (basic single track).

the number of priority levels. This will increase the length of the write phase and result in longer cell processing time.

The main drawback of the switch is the sequential processing of the token by input and output ports. As a result, the time taken for the token to travel through the entire ring increases as the size of the switch increases. In case of multiple priority levels, the recirculation period for the token becomes even longer. Here, head-of-line (HOL) blocking is another factor that degrades throughput.

15.1.3 Cisneros and Brackett

Cisneros and Brackett [11] proposed a large ATM switch architecture based on memory switch modules and optical star couplers. The architecture consists of input modules, output modules, optical star couplers, and a contention resolution device (CRD). Input and output modules are based on electronic shared memories. The architecture requires optical to electronic and electronic to optical conversions in some stages. Each output module has an associated unique wavelength. As shown in Figure 15.3, the input ports and output ports are grouped into size n, and each group is connected to $n \times m$ and $m \times n$ memory switches, respectively. The interconnection between the input and output modules is achieved by k optical star couplers. There are k tunable laser transmitters and k fixed wavelength receivers connected to each optical star coupler. (In Fig. 15.3, optical transmitters and receivers are not shown in order to keep it simple). The cells transmitted through the switch are buffered at the input and output modules. In the proposed architecture, input and output lines transmit cells at the rate of 155.52 Mbit/s. The lines that interconnect the input modules to the optical stars and optical stars to the output modules run at 2.5 Gbit/s. The values of n, k, N, and m are 128, 128, 16,384, and 8, respectively.

The internal routing header of a cell is composed of two fields. One specifies the output module and the other shows the port number in that specific output module. Each input

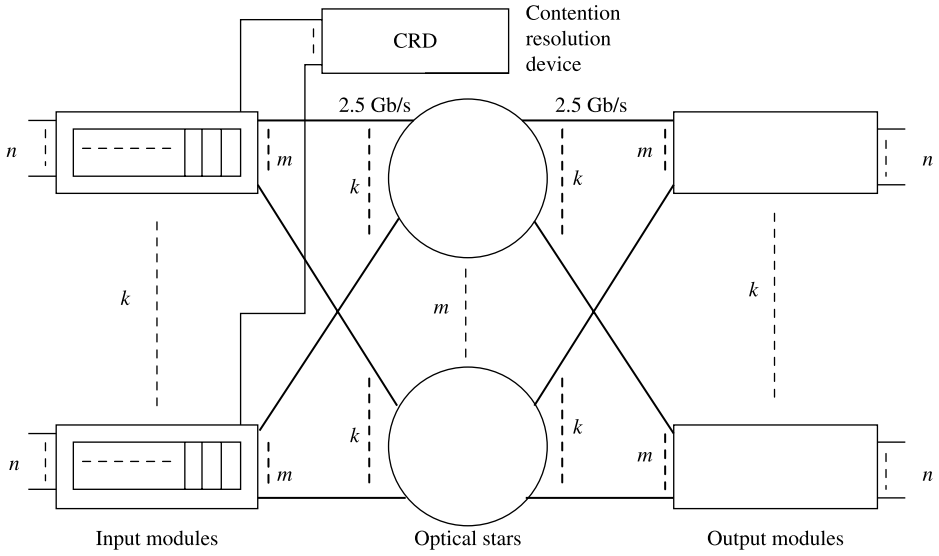


Figure 15.3 Switch architecture proposed by Cisneros and Bracket (© 1991 IEEE).

module handles a single queue in which the incoming cells are kept in sequence. The input modules, the optical stars, and the output modules are connected as in a three-stage Clos network. However, the working principle is not the same as in the Clos network. Here, each input module sends the output module request of its HOL cell to the CRD. The CRD examines the requests and chooses one cell for each output module and responds. The cells that won the contention are routed through the first $k \times k$ optical star and their HOL pointers are advanced. This process is repeated for each optical star in a circular manner. Cells at the output modules are kept in sequence depending on which optical star they arrive in. In this architecture, all optical stars are kept busy if the CRD is m times faster than a cell transfer time by the optical stars. The maximum amount of optical couplers is determined with respect to the time required to transfer a cell through an optical star and the time required to resolve contention by the CRD.

In the architecture, the time required for optical to electronic conversion, electronic to optical conversion, and tuning optical laser transmitters is not considered. All the calculations are mainly based on the time required to transfer a cell through an optical star. The output port contention resolution scheme is very complex and the electronic controller can become a bottleneck. The switch does not have multicast capability due to the fixed wavelength receivers. Moreover, the maximum throughput of the switch is limited to 58 percent because of the HOL blocking [12].

15.1.4 BNR (Bell-North Research) Switch

Munter et al. [13] introduced a high-capacity packet switch based on advanced electronic and optical technologies. The main components of the switch are input buffer modules, output buffer modules, a high-speed switching core, and a central control unit, as shown in Figure 15.4. The core switch contains a 16×16 cross-connect network using optical links running at 10 Gbit/s.

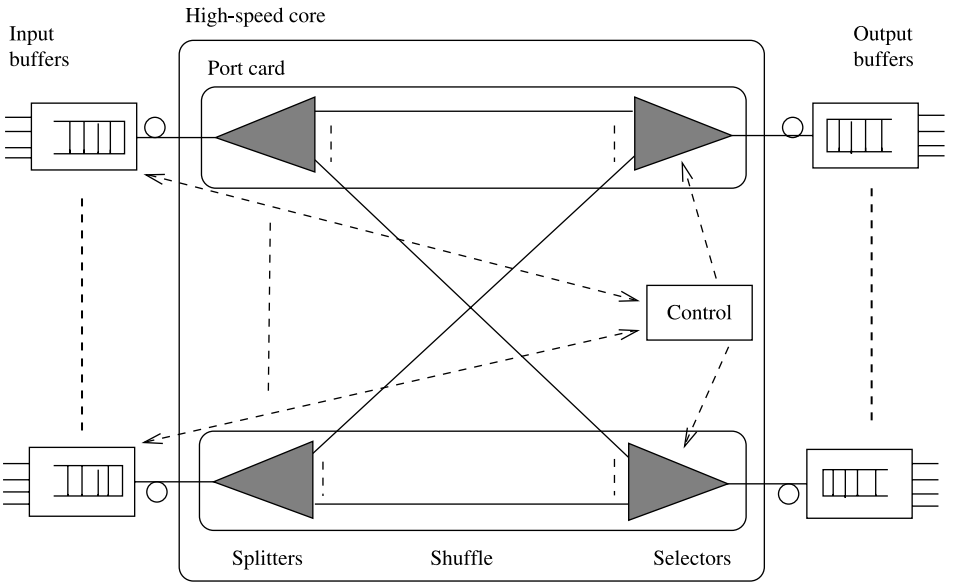


Figure 15.4 Diagram of BNR switch.

The central control unit receives requests from input buffer modules and returns grant messages. Each request message indicates the number of queued packets in the input buffer module, which is later used to determine the size of burst allowed to transmit to the switch fabric. A connection can only be made when both input and output ports are free. A control bus is used by the free input ports to broadcast their requests, and by the free output ports to return grant messages.

An arbitration frame consists of 16 packet time slots for a 16×16 core switch. In each slot, the corresponding output port polls all 16 inputs. For example, in time slot 1, output port 1 (if it is idle) will choose the input that has the longest queue destined for output port 1. If the input is busy, another input port that has the second longest queue will be examined. This operation repeats until a free input port is found. If a match is found (free input, free output, and outstanding request), a connection is made for the duration corresponding to the number of packets queued for this connection. So, the switch is a burst switch, not a packet switch. In time slot 2, output port 2 repeats the above operation. The switch capacity is limited by the speed of the central control unit. Packet streams can have a long waiting time in the input buffer modules under a high traffic load.

15.1.5 Wave-Mux Switch

Nakahira et al. [14] introduced a photonic asynchronous transfer mode (ATM) switch based on the input–output buffering principle. Basically, this switch consists of three kinds of modules: input group module (IGM), switching module (SWM), and output group module (OGM), as shown in Figure 15.5. They are connected by means of fiber optical lines. The inputs are divided into p groups of size n_1 and each group is connected to an IGM. The cells arriving through optical lines are first converted to electronic signals by optical-to-electrical (O/E) converters and their header information is electrically processed at the header converter in IGMs. Both the header and payload of the arriving cell are processed and stored in an electronic random access memory (RAM). An optical sorter in each IGM is used to sort the cells with respect to their OGM requests and delivers them to SWM in a cell slot time.

There are p optical switches in SWM. Each optical switch transmits optical wavelength multiplexed cells from IGM to OGM. In each cell time slot, these p optical switches deliver

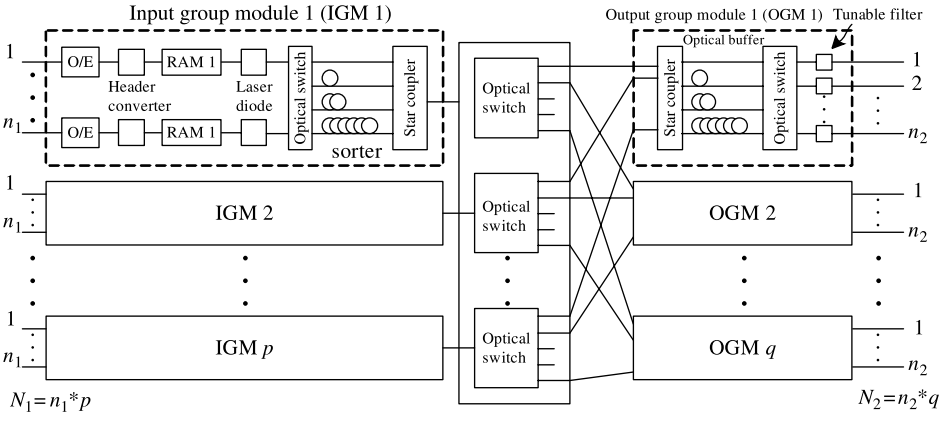


Figure 15.5 Architecture of the wave-mux switch (© 1995 IEEE).

at most p trunks of wavelength multiplexed cells from the IGMs, which are destined for different OGMs. In each OGM, it is possible to have the cells with different wavelengths but with the same output port request. This is called output port contention and is solved by the use of an optical buffer. This optical buffer in the OGM is based on the fiber delay line principle. If no competing cells to the same output port are present in the optical buffer, the incoming wavelengths will be sent through the shortest optical fiber line. They are distributed to the tunable filters by an optical switch. Each tunable filter is then tuned to the wavelength of the requested output port.

The proposed optical switch architecture needs complex arbitration to solve the contention problem, not only for those cells in same IGM but for the cells in different IGMs as well. This will increase the complexity of control, thus limiting the switch size. In this switch, in order to avoid HOL blocking, cells destined for the same OGM can be read out of the RAM with the speed-up factor by two. There are many O/E and E/O converters required in the switching path, thus increasing implementation costs.

15.2 OPTOELECTRONIC PACKET SWITCH CASE STUDY I

Figure 15.6 shows a terabit IP router architecture with four major elements in the terabit IP router [15]: the optical interconnection network (OIN) supporting nonblocking and high-capacity switching, the ping-pong arbitration unit (PAU) resolving the output contention and controlling the switching devices, the router modules (RMs) performing IP packet forwarding, and the route controller (RC) constructing routing information for the RMs. There are two kinds of RM: input RM (IRM) and output RM (ORM). Both the IRMs and the ORMs implement IP packet buffering, route (table) lookup, packet filtering, and versatile interfaces, such as OC-3, OC12, OC-48, and Gigabit Ethernet. The interconnection between the RC and the RMs can be implemented with dedicated buses or through the OIN. Figure 15.6 simply illustrates the bus-based approach.

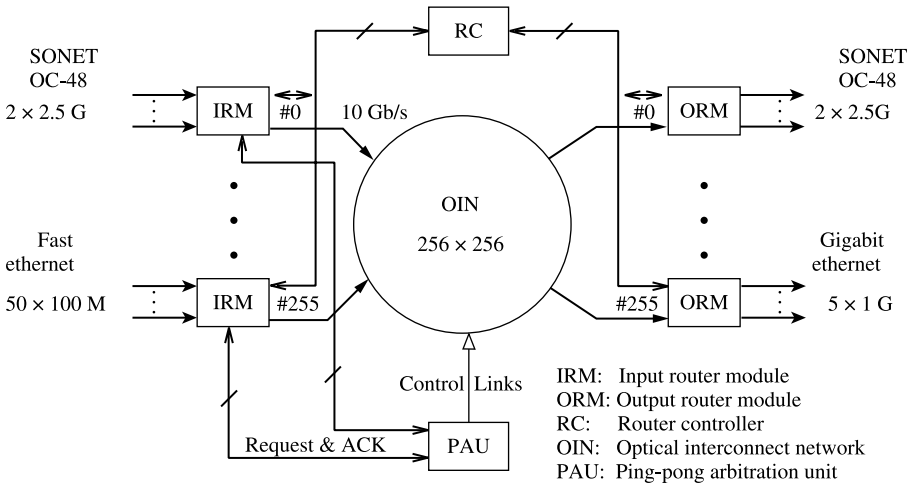


Figure 15.6 Architecture of a terabit IP router (© 1998 IEEE).

15.2.1 Speedup

The fixed-length segment switching technique is commonly adopted in high-capacity IP routers to achieve high-speed switching and better system performance.

Figure 15.7*a* suggests that a speedup factor of two is required to achieve nearly 100 percent throughput under bursty traffic with geometric distribution and an average burst size of 10 packet segments. Figure 15.7*b* shows the corresponding average delay. The total average delay of input and output queuing is very close to the theoretic bound of purely

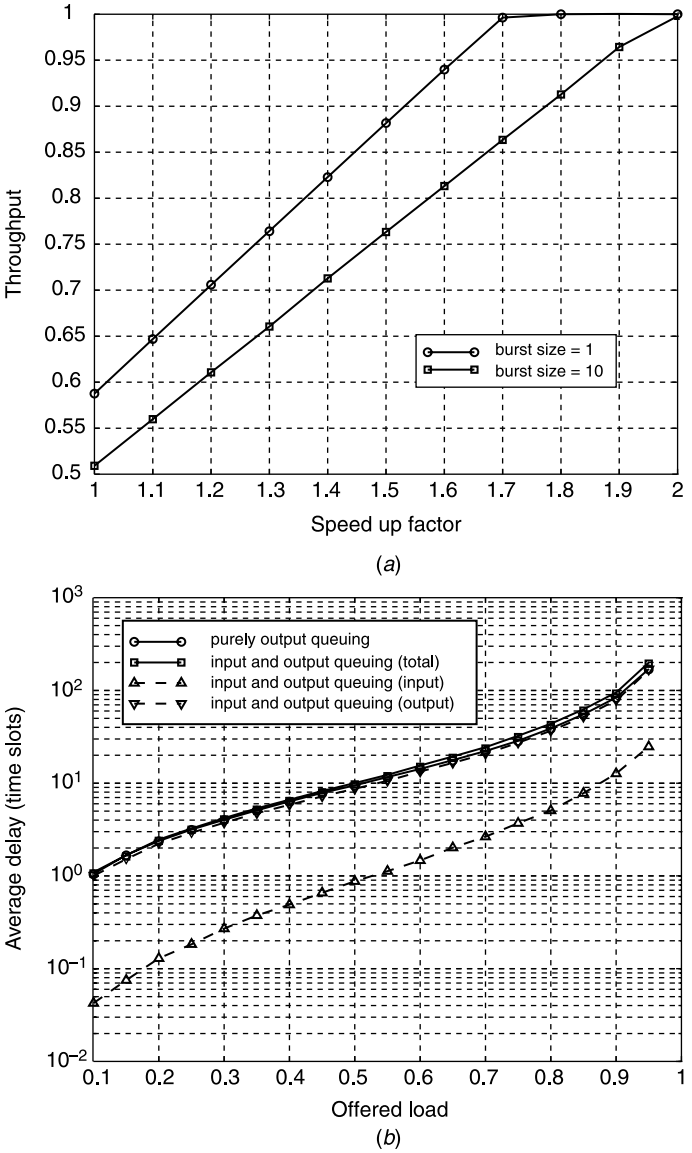


Figure 15.7 Switch performance: (*a*) Throughput; (*b*) Average delay with burst size = 10 and speedup factor = 2.

output queuing. The input delay is an order smaller than the total delay, hinting that an input queued switch with speedup 2, in the average sense, performs nearly as well as a purely output queued switch.

The speedup induces two more challenges: (1) doubling the switch transmission speed to 10 Gbit/s, and (2) halving the arbitration time constraint. The first challenge can be easily resolved with optical interconnection technology, while the second challenge can be resolved by the ping-pong arbitration (PPA) scheme described in Section 15.2.4.

15.2.2 Data Packet Flow

A data segment unit of 64 bytes is chosen to accommodate the shortest IP packets (40 bytes). Variable-length IP packets are segmented before being passed through the switch. Figure 15.8 depicts the flow of packets across the router. A simple round-robin packet scheduler is used at each input line interface (ILI) to arrange the packet arrivals from different interfaces (see also Fig. 15.6). It uses a first-in-first-out (FIFO) buffer per interface to store incoming packets. Since the output line speed of the scheduler is the sum of all interfaces, it can be shown that the maximum packet backlog at each input line FIFO is just twice that of the maximum IP packet size, the same large buffer can be chosen to avoid any packet loss.

The output packets of the scheduler enter the input switch interface (ISI) in which packet segmentation takes place. While a packet is being segmented, its IP header is first checked by the input packet filter (IPF) for network security and flow classification (i.e., inbound filtering), as shown in Figure 15.6. Afterwards, the header is sent to the input forwarding engine (IFE) for IP table lookup, deciding which ORM(s) the packet is destined for.

Data segments are stored in a FIFO waiting for arbitration before being forwarded through the OIN. The forwarding sequence is packet-by-packet, not cell-by-cell, for each ISI in order to simplify the reassembly. The input port number is added to each segment before it enters the OIN to ensure correct packet reassembly at the output ports.

Segments of a packet arriving at an output port may be interleaved with those from other input ports. While a packet is being reassembled, its IP header can be sent to the output packet filter (OPF) for outbound filtering and then to the output forwarding engine (OFE) for another IP route lookup deciding which outgoing interface(s) the packet should be destined for. The packets are then broadcast at the output line interface (OLI) to all desirable interfaces. Each interface can maintain two FIFOs supporting two priority traffic: real-time (RT) and non-real-time (NRT) packets.

15.2.3 Optical Interconnection Network (OIN)

Figure 15.9 shows the proposed 256×256 OIN, which can easily provide the multicast function due to its broadcast-and-select property. The OIN consists of two kinds of optical switching modules: input optical modules (IOMs) and output optical modules (OOMs). There are 16 of each kind in the OIN. Each IOM uses the same set of 16 different wavelengths (λ_1–λ_{16}); each of the 16 input links at an IOM is assigned a distinct wavelength from the set, which carries packet segments under transmission. In each time slot, up to 16 packet segments at an IOM can be multiplexed by an arrayed-waveguide grating (AWG) router. The multiplexed signal is then broadcast to all 16 OOMs by a passive 1×16 splitter.

At each OOM, a 16×16 fully connected switching fabric performs the multicast switching function by properly controlling the semiconductor optical amplifier (SOA) gates. There

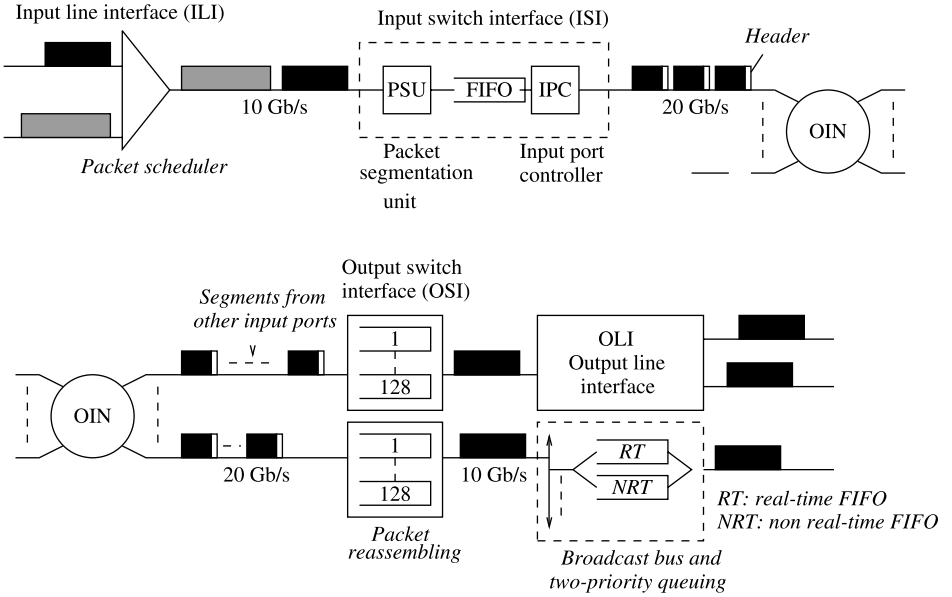


Figure 15.8 Flow of packets across the router.

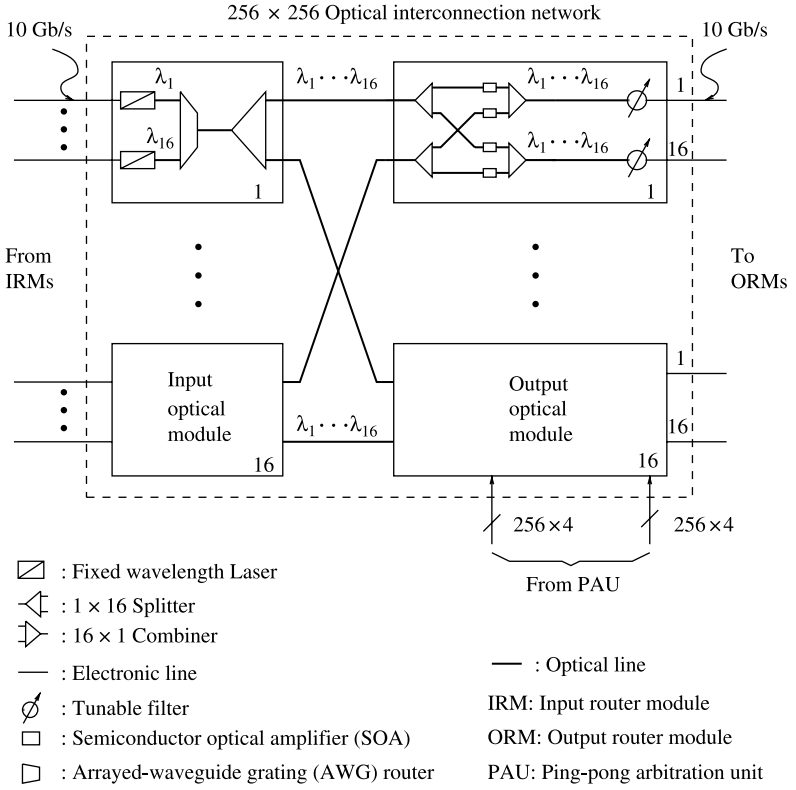


Figure 15.9 256 × 256 OIN.

are a total of 256 SOA gates in each OOM. At most 16 of them can be turned ON simultaneously. The tunable filter, controlled by the PAU, is used to dynamically choose one of the 16 wavelengths in every time slot. As illustrated in Figure 15.10, it is assumed that a packet segment from the kth input link of the ith IOM is destined for the qth and the 16th output links of the jth OOM, where $1 \leq i, j, k, q \leq 16$. These two multicast connections are established by turning on the SOA gates with index (i, j, q) and $(i, j, 16)$ only (the others are turned off). The tunable filters at the qth and the 16th output links of the jth OOM are turned on with index k, which is provided by the PAU.

Input Optical Module (IOM). The IOMs carry packets at 10 Gbit/s. At each IOM, distributed Bragg reflector (DBR) or distributed feedback (DFB) laser arrays can be used as the laser sources between 1525 nm and 1565 nm to match the gain bandwidth of commercially available erbium-doped fiber amplifiers (EDFAs). Each EDFA can amplify multiple wavelengths simultaneously. Each input link of an IOM is connected to a laser with fixed wavelength.

To improve the chirp performance, a DFB laser diode integrated with an external modulator (EM) operating at 10 Gbit/s has been fabricated [16]. To ensure output power levels and chirp performance, a SOA and EMs can be integrated with the DFB laser arrays [17]. This monolithically integrated WDM source is able to provide multi-wavelength capability and significantly reduce the cost per wavelength. In addition, it can also eliminate the

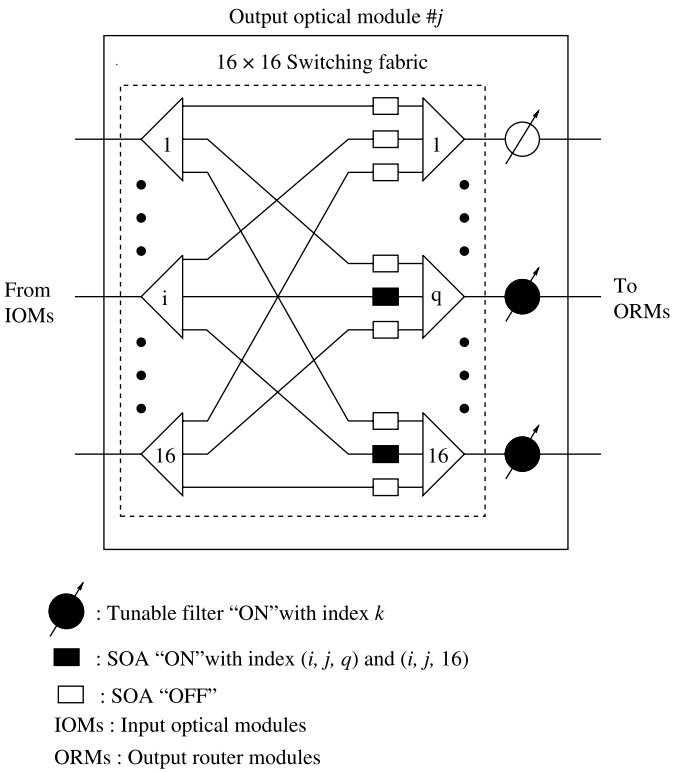


Figure 15.10 Control in the jth output optical module (OOM).

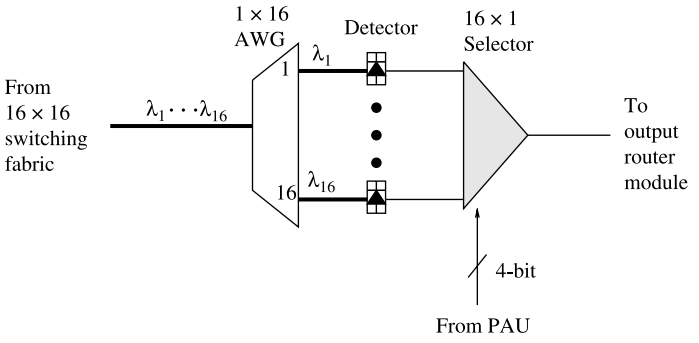
alignment of fibers to individual lasers, reduce component count and coupling loss between components, and increase the reliability.

Output Optical Module (OOM). Each 16 × 16 switching fabric should be capable of simultaneously connecting two or more IOMs to all tunable filters at an OOM. Thus, it needs to have broadcast capability and to be strictly nonblocking. As shown in Figure 15.10, a space switch can simply meet this requirement and can be constructed by using SOA gates.

In addition to their fast switching function (∼1 ns), SOA gates can provide some gain to compensate the coupling loss and splitting loss caused by the splitters/combiners and the connection between discrete optical devices. Furthermore, SOA gates can be monolithically integrated with the passive couplers to enhance the reliability and loss performance between components.

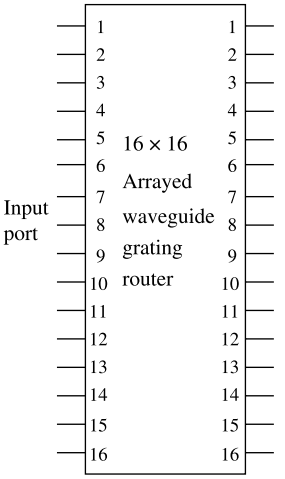
Tunable Filters. Tunable filters are used to perform wavelength selection in the OIN. Three possible ways to implement the tunable filter are considered here.

Type-I Tunable Filter. A Type-I tunable filter, as shown in Figure 15.11, performs the wavelength selection in the electrical domain. Each output of a 16 × 16 switching fabric is connected to a 1 × 16 AWG router, which is made from high-index indium phosphide (InP) material and is capable of demultiplexing 16 wavelengths in the 1550 nm window. Figure 15.12 shows the connectivity of a 16 × 16 AWG router. For example, if the WDM signal enters the seventh input port of the AWG router, only the 14th wavelength (λ_{14}) will be sent out through the eighth output port. Each demultiplexed wavelength is detected through a high-speed signal detector. Each detector has a laser waveguide structure and can be monolithically integrated with the AWG router, thus increasing the reliability and reducing the packaging cost of the AWG router. Finally, a 16 × 1 electronic selector is used to select the desired signal from the 16 detectors. The selector is controlled by the 4-bit control signal from the PAU. An alternative electronic selector is an InP-based optoelectronic integrated circuits (OEIC) receiver array [18], which operates at 10 Gbit/s per channel and integrates 16 p-i-n photodiodes with heterojunction bipolar transistors (HBT) preamplifier.



AWG : Arrayed-waveguide grating
PAU: Ping-pong arbitration unit

Figure 15.11 Type-I tunable filter.

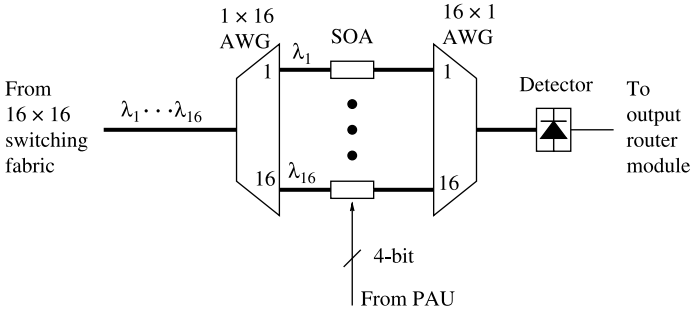


Input port	Output port															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	1
3	3	4	5	6	7	8	9	10	11	12	13	14	15	16	1	2
4	4	5	6	7	8	9	10	11	12	13	14	15	16	1	2	3
5	5	6	7	8	9	10	11	12	13	14	15	16	1	2	3	4
6	6	7	8	9	10	11	12	13	14	15	16	1	2	3	4	5
7	7	8	9	10	11	12	13	14	15	16	1	2	3	4	5	6
8	8	9	10	11	12	13	14	15	16	1	2	3	4	5	6	7
9	9	10	11	12	13	14	15	16	1	2	3	4	5	6	7	8
10	10	11	12	13	14	15	16	1	2	3	4	5	6	7	8	9
11	11	12	13	14	15	16	1	2	3	4	5	6	7	8	9	10
12	12	13	14	15	16	1	2	3	4	5	6	7	8	9	10	11
13	13	14	15	16	1	2	3	4	5	6	7	8	9	10	11	12
14	14	15	16	1	2	3	4	5	6	7	8	9	10	11	12	13
15	15	16	1	2	3	4	5	6	7	8	9	10	11	12	13	14
16	16	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Figure 15.12 16 × 16 arrayed-waveguide grating (AWG) router connectivity.

Type-II Tunable Filter. A Type-II tunable filter, as shown in Figure 15.13, performs the wavelength selection optically. It has two AWGs. The first 1 × 16 AWG performs the demultiplexing function, while the second 16 × 1 AWG performs the multiplexing function. Only one of 16 wavelengths is selected by properly controlling the SOA gates. The selected wavelength passes through the second AWG and is then converted into an electronic signal by a detector. A planar lightwave circuit–planar lightwave circuit (PLC–PLC) direct attachment technique [19] can be used to construct this type of tunable filter and to integrate the AWG routers and the SOA gates. This hybrid integration of PLC and SOA gates can reduce the coupling loss and increase the reliability.

Type-III Tunable Filter. A Type-III tunable filter, as shown in Figure 15.14, performs the wavelength selection optically. Different from the Type-II filter, it uses only one 16 × 16 AWG router. Any one of the 16 wavelengths can be selected through its specific combination of SOA gates at input and output sides of AWG router [20]. Figure 15.15 shows a way to



PAU: Ping-pong arbitration unit
AWG: Arrayed-waveguide grating
SOA: Semiconductor optical amplifier

Figure 15.13 Type-II tunable filter.

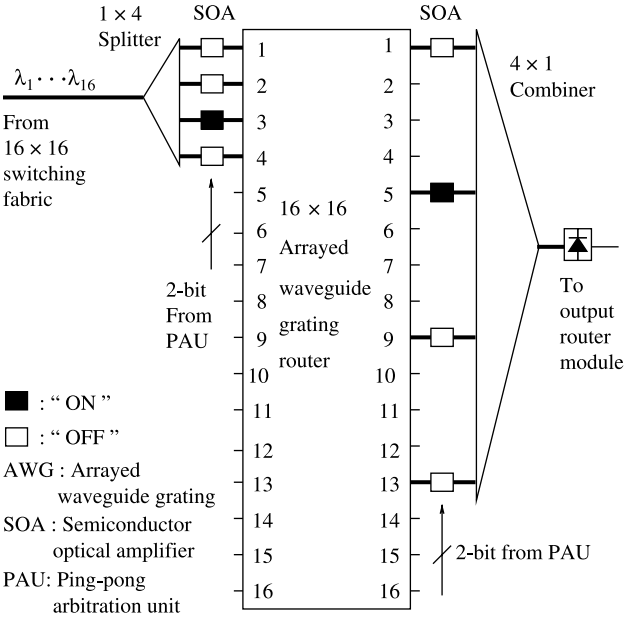


Figure 15.14 Type-III tunable filter.

choose any one of the 16 wavelengths. The 16×16 AWG router will route a wavelength λ_k from input port x ($x = 1, 2, \ldots, 16$) to output port y ($y = 1, 2, \ldots, 16$), where $k = (x + y - 1)$ modulo 16. For example, λ_7 will be selected as the output by turning on the third SOA gate at the input side and the fifth SOA gate at the output side of the AWG router, respectively. The quantity of SOA gates in the Type-III tunable filter is reduced by half; only eight SOA gates (four at the input and four at the output) are used instead of 16 SOA gates in the Type-II tunable filter. However, compared to Type-I and Type-II tunable filters, the Type-III tunable filter has more power loss caused by the 1×4 splitter and the 4×1 combiner.

15.2.4 Ping-Pong Arbitration Unit

As shown in Figure 15.6, a centralized PAU was used in the router [15]. The arbitration is pipelined with packet segment transmission in the OIN. In other words, while a HOL segment is being transmitted via the OIN, the segment next to it is also sending a request

Input port	Output port			
	1	5	9	13
1	1	5	9	13
2	2	6	10	14
3	3	7	11	15
4	4	8	12	16

Figure 15.15 Connectivity of Type-III tunable filter.

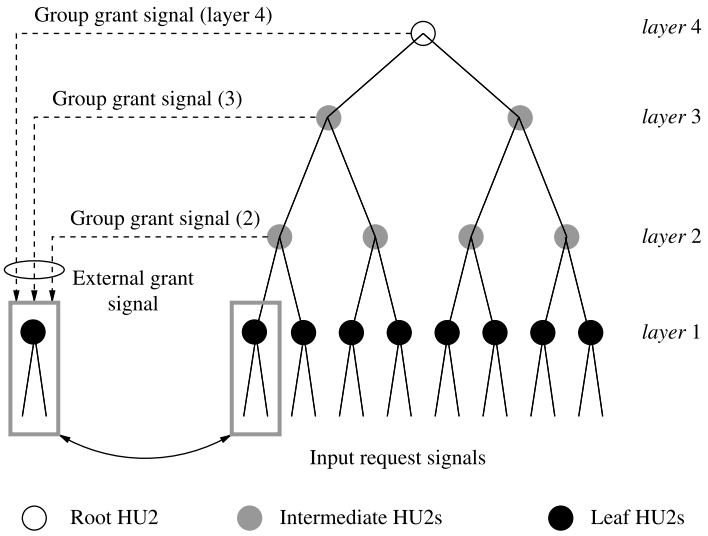


Figure 15.16 Tree-structured hierarchical arbitration (© 1999 IEEE).

to the arbitration unit. In order to minimize the delay of forwarding multicast request signals, 256 parallel arbiters are used, each of which is associated with one output and handles 256 input request signals. The 256 incoming multicast request signals must be handled simultaneously within one time slot, that is, 51.2 ns for 64-byte data segment sent at 10 Gbit/s.

Principles of PPA. Consider an N-input packet switch. To resolve its output contention, a solution is to use an arbiter for each output to fairly select one among those incoming packets and send back a grant signal to the corresponding input. The arbitration procedure is as follows:

1. During every arbitration cycle, each input submits a one-bit request signal to each output (arbiter), indicating whether its packet, if any, is destined for the output.
2. Each output arbiter collects N request signals, among which one input with active request is granted according to some priority order.
3. A grant signal is sent back to acknowledge the input.

Here, the second step that arbitrates one input among N possible ones is considered.

A simple round robin scheme is generally adopted in an arbiter to ensure a fair arbitration among the inputs. Imagine there is a token circulating among the inputs in a certain ordering. The input that is granted by the arbiter is said to grasp the token, which represents the grant signal. The arbiter is responsible for moving the token among the inputs that have request signals. The traditional arbiters handle all inputs together and the arbitration time is proportional to the number of inputs. As a result, the switch size or capacity is limited given a fixed amount of arbitration time.

Here, it is suggested to divide the inputs into groups with each group having its own arbiter. The request information of each group is summarized as a group request signal. Further grouping can be applied recursively to all the group request signals at the current

layer, forming a tree structure, as illustrated in Figure 15.16. Thus, an arbiter with N inputs can be constructed using multiple small-size arbiters (AR) at each layer. Different group sizes can be used.

Assume $N = 2^k$. Figure 15.16 depicts a k-layer complete binary tree with a group size of two when $k = 4$. AR2 represents a 2-input AR. An AR2 contains an internal flag signal that indicates which input is favored. Once an input is granted in an arbitration cycle, the other input will be favored in the next cycle. In other words, the granted request is always chosen between left (input) and right alternately. That is why it is called ping-pong arbitration. The first layer consists of 2^{k-1} arbiters and are called leaf AR2s. The next $k - 2$ layers consist of arbiters called intermediate AR2s, 2^{k-i} of which are at layer i. Finally, the last layer consists of only one arbiter called a *root* AR2.

Every AR2 has two request signals. An input request signal at layer i is the group request signal of 2^{i-1} inputs and can be produced by OR gates, either directly or recursively. The grant signal from an AR2 has to be fed back to all the lower-layer AR2s related to the corresponding input. Therefore, every leaf/intermediate AR2 also has an external grant signal that ANDs all grant signals at upper layers, indicating the arbitration results of upper layers. The root AR2 needs no external grant signal. At each leaf AR2, the local grant signals have to combine the upper-layer arbitration results (i.e., its external grant signal) and provide full information of whether the corresponding input is granted or not.

One important usage of the external grant signal is to govern the local flag signal update. If the external grant signal is invalid, which indicates that these two input requests as a whole are not granted at some upper layer(s), then the flag should be kept unchanged in order to preserve the original preference. As shown in Figure 15.16, the external grant signal of a leaf AR2 can be added at the final stage to allow other local logical operations to be finished while waiting for the grant signals from upper layers, which minimizes the total arbitration time.

Suppose N inputs are served in the increasing order of their input numbers, that is, $1 \rightarrow 2 \rightarrow \cdots \rightarrow N \rightarrow 1$ under a round-robin scheme. Each AR2 by itself performs a round-robin service for its two inputs. The PPA, consisting of a tree of AR2s, is shown in Figure 15.16. It can serve the inputs in the order of $1 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 1$ when $N = 4$ for instance, which is still round-robin, if each input always has a packet to send and there is no conflict between all the input request signals. Its performance is shown by simulations as follows.

Performance of PPA. The performance of the PPA, FIFO + RR (FIFO for input queuing and round robin for arbitration), and output queuing is compared here. A speedup factor of two is used for PPA and FIFO + RR. Simulation results are obtained from a 32×32 switch under uniform traffic (the output address of each segment is equally distributed among all outputs), and bursty traffic (on–off geometric distribution) with an average burst length of 10 segments. The bursty traffic can be used as a packet traffic model with each burst representing a packet of multiple segments destined for the same output. The output address of each packet (burst) is also equally distributed among all outputs.

Figure 15.17 shows the throughput and total average delay of the switch under various arbitration schemes. It can be seen that the PPA performs comparably with the output queuing and the FIFO + RR. However, the output queuing is not scalable and the RR arbitration is slower than the PPA. The overall arbitration time of the PPA for an N-input switch is proportional to $\log_4 \lceil N/2 \rceil$ when every four inputs are grouped at each layer. For instance, the PPA can reduce the arbitration time of a 256×256 switch to 11 gates delay, less than 5 ns using the current CMOS technology.

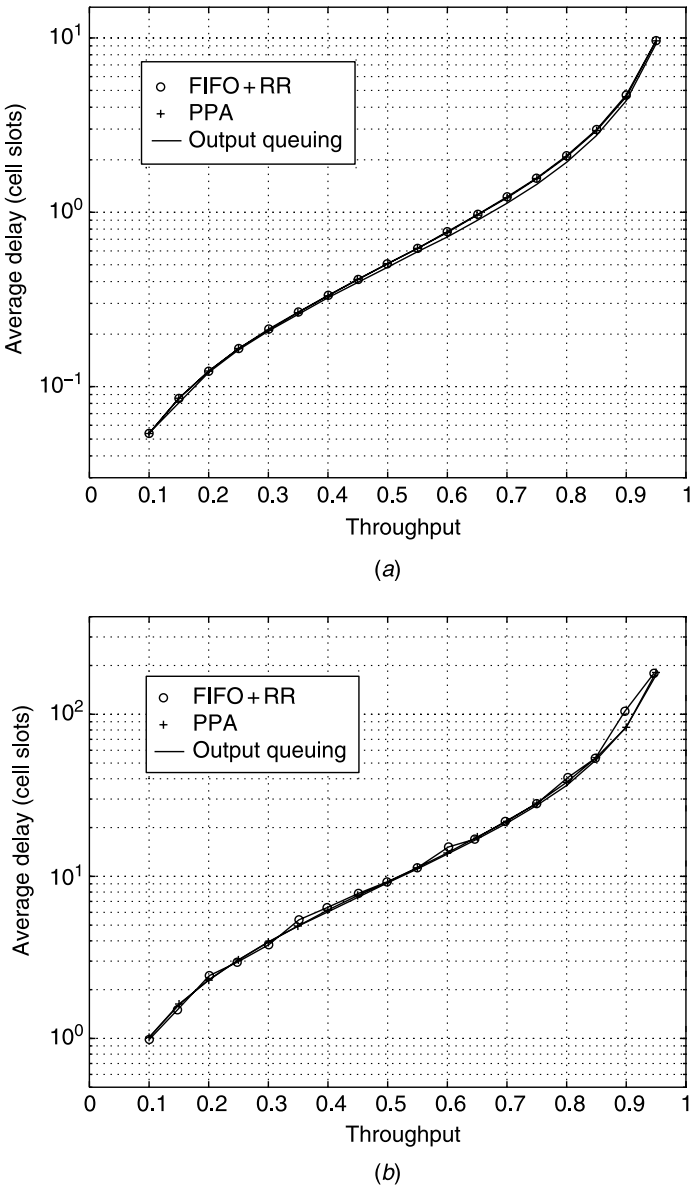


Figure 15.17 Comparison of the PPA with FIFO + RR and output queuing: switch throughput and total average delay for a speedup factor of two. (*a*) Uniform traffic; (*b*) Bursty traffic.

Implementation of PPA. Multiple small arbiters can be recursively grouped together to form a large and multi-layer arbiter, as illustrated in Figure 15.16. Figure 15.18 depicts an n-input arbiter constructed by using p q-input arbiters (AR-q), from which the group request/grant signals are incorporated into a p-input arbiter (AR-p). Constructing a 256-input arbiter starting with the basic units, AR2s, is shown as follows.

Figure 15.19 shows a basic 2-input arbiter (AR2) and its logical circuits. The AR2 contains an internally feedbacked flag signal, denoted by F_i, that indicates which input is

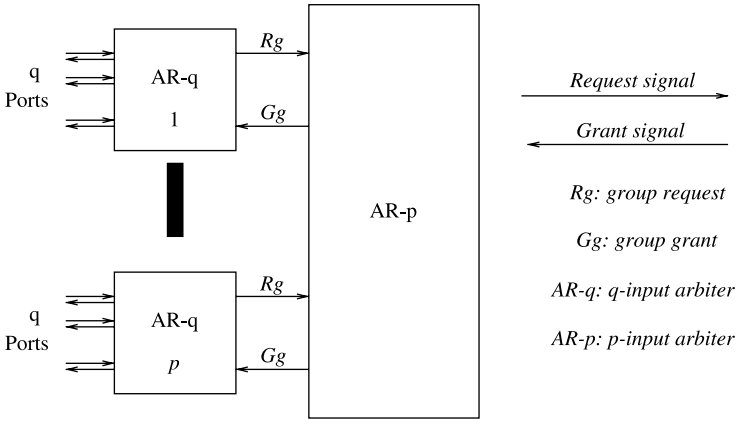
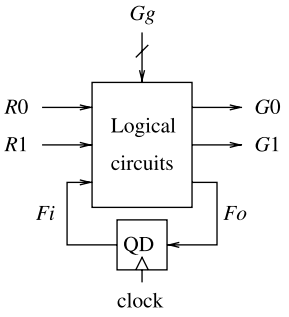


Figure 15.18 Hierarchy of recursive arbitration with $n = pq$ inputs.



*R*0 : *Request* 0
*R*1 : *Request* 1

*G*0 : *Grant signal* 0
*G*1 : *Grant signal* 1

Fi : *Flag input*
Fo : *Flag output*

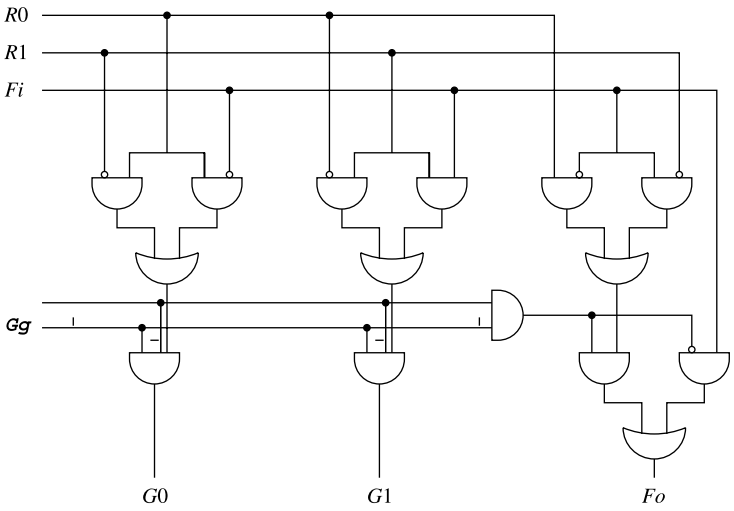
Gg : *Group grant*

True table

Input			Output		
*R*0	*R*1	*Fi*	*G*0	*G*1	*Fo*
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	0	1
1	1	1	0	1	0

(when all Gg inputs are 1)

(*a*)



(*b*)

Figure 15.19 (*a*) AR2 and its true table; (*b*) its logical circuits (© 1999 IEEE).

favored.[1] When all G_g inputs are 1, indicating these two inputs requests (R_0 and R_1) as a whole are granted by all the upper layers, once an input is granted in an arbitration cycle, the other input will be favored in the next cycle, as shown by the true table in Figure 15.19*a*. This mechanism is maintained by producing an output flag signal, denoted by F_o, feedbacked to the input. Between F_o and F_i, there is a D-flip-flop that functions as a register forwarding F_0 to F_i at the beginning of each cell time slot. When at least one G_g inputs is 0, indicating the group request of R_0 and R_1 is not granted at some upper layer(s), $G_0 = G_1 = 0$, $F_o = F_i$, that is, the flag is kept unchanged in order to preserve the original preference. As shown in Figure 15.19*b*, the local grant signals have to be ANDed with the grant signals from the upper layers to provide full information on whether the corresponding input is granted or not. G_g inputs are added at the final stage to allow other local logical operations to be finished in order to minimize the total arbitration time.

A 4-input arbiter (AR4) module has four request signals, four output grant signals, one outgoing group request and one incoming group grant signal. Figure 15.20*a* depicts our design of an AR4 constructed by three AR2s (two leaf AR2s and one intermediate AR2; all have the same circuitry), two 2-input OR gates and one 4-input OR gate. Each leaf AR2 handles a pair of inputs and generates the local grant signals while allowing two external grant signals coming from the upper layers: one from the intermediate AR2 inside the AR4 and the other from outside AR4. These two signals directly join the AND gates at the final stage inside each leaf AR2 for minimizing the delay. Denote R_{ij} and G_{ij} as the group request signal and the group grant signal between input i and input j. The intermediate AR2 handles the group requests (R_{01} and R_{23}) and generates the grant signals (G_{01} and G_{23}) to each leaf AR2, respectively. It contains only one grant signal that is from the upper layer for controlling the flag signal.

As shown in Figure 15.20*b*, 16-input arbiter (AR16) contains five AR4s in two layers: four at the lower layer handling the local input request signals and one at the higher layer handling the group request signals.

Figure 15.21 illustrates a 256-input arbiter (AR256) constructed using AR4s and its arbitration delay components. The path numbered from 1 to 11 shows the delay from the point when an input sends its request signal up until it receives the grant signal. The first four gate delays (1–4) account for the time taken for the input's request signal to pass though the four layers of AR4s and reach the root AR2, where one OR-gate delay is needed at each layer to generate the request signal (see Fig. 15.20*a*). The next three gate delays (5–7) account for the time that the root AR2 performs its arbitration (see Fig. 15.19*b*). The last four gate delays (8–11) account for the time that the grant signals at the upper layers take to pass down to the corresponding input. The total arbitration time of an AR256 is thus 11 gates delay. It then follows that the arbitration time (T_n) of an n-input arbiter using such implementation is

$$T_n = 2\log_4 \left\lceil \frac{n}{2} \right\rceil + 3. \tag{15.1}$$

Priority PPA. Among the packets contending for the same output, those from real-time sessions are more delay-sensitive than others from non-real-time sessions. Therefore, they should have a higher priority to be served first, and sessions (thus their packets) with various

[1]When the flag is low, R_0 is favored; when the flag is high, R_1 is favored.

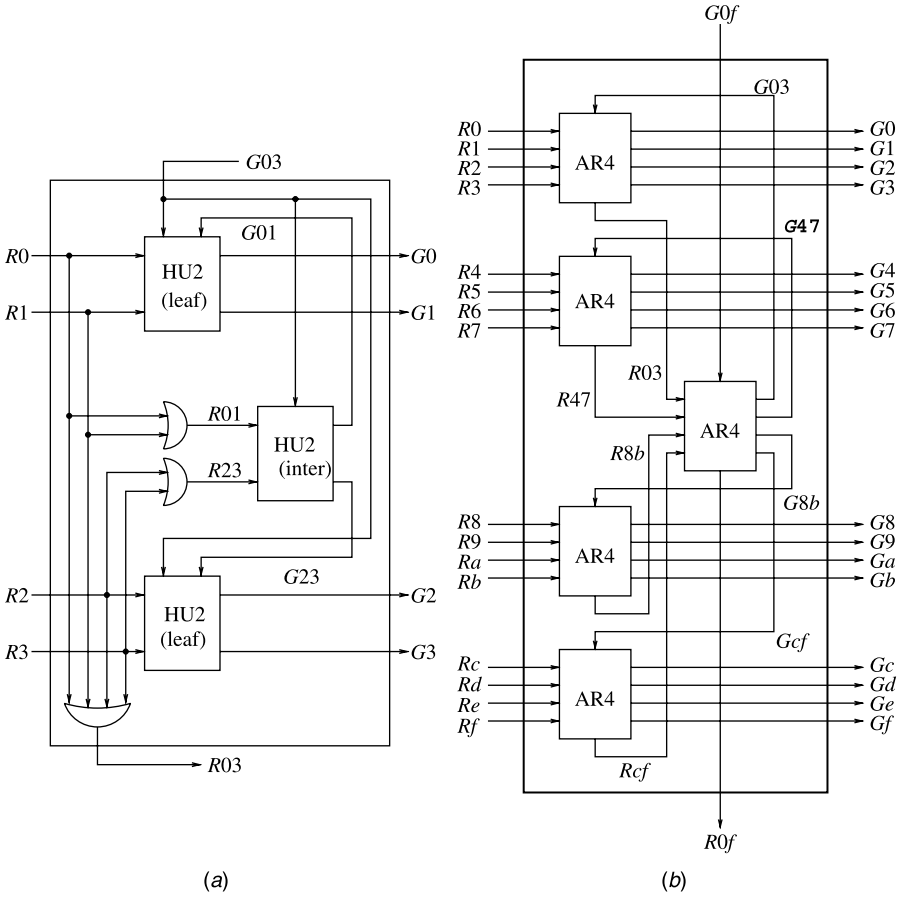


Figure 15.20 (*a*) AR4; (*b*) AR16 constructed with five AR4s (© 1999 IEEE).

quality of service (QoS) requirements can be assigned different levels of service priority. It is shown how to enhance the PPA for handling priority as follows.

Two priority representations are used in our design for transferring efficiency and arbitration convenience, respectively. Suppose p levels of priority are supported. An input has a total of $p + 1$ states, including the case of no request, which can be represented by using $\lceil \log_2(p + 1) \rceil$ bits. The inter-layer request information could be transferred either in serial using one line or in parallel using multiple lines, depending on the tradeoff chosen between delay and pin count complexity. The serial/parallel format transformation can be realized by using shift registers.

A group of p lines is used in the second representation. At most, one of them is high indicating that there is one request at the corresponding level of priority. There will be no request if all output lines are low.

Our solution to multi-priority arbitration relies in a group of parallel single-priority arbiters to resolve the contention at each level of priority simultaneously. Multiple single-priority arbiters are necessary to maintain the arbitration states (states of the flip-flops) for each level of priority, which will be changed only when an input request at this priority

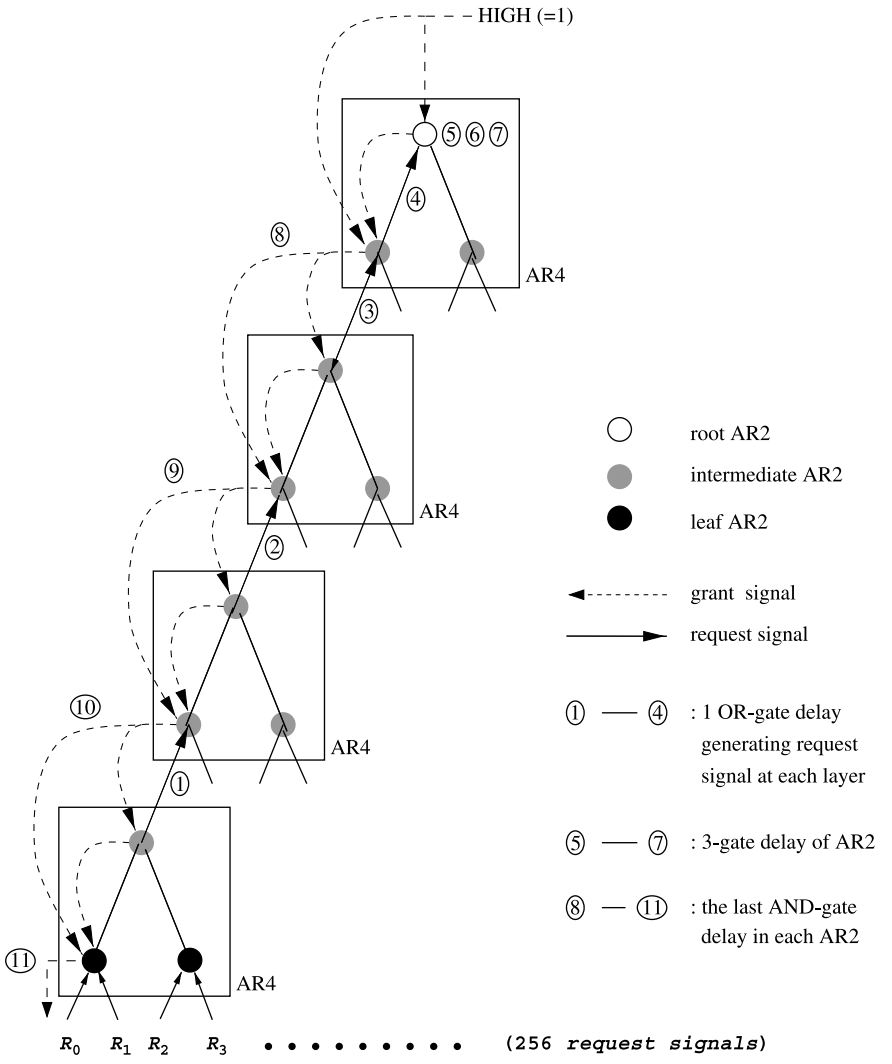


Figure 15.21 Decomposition of arbitration delay in an AR256 (© 1999 IEEE).

level is granted. A pre-processing phase and a post-processing phase are then added, as demonstrated in Figure 15.22, with a multi-priority arbiter, which handles 16 inputs and seven levels of priority. A decoder is used at each input to decode the three-line priority request into seven single lines, each representing the request in the corresponding level of priority and entering the corresponding arbiter for single-priority contention resolution. An OR gate is used at each output to combine all corresponding local grants from the single-priority arbiters to produce the final grants for each input.

Meanwhile, every single-priority arbiter generates a group request signal for the upper layer's arbitration; it receives a group grant signal later, which indicates if this group of requests (at the corresponding level of priority) is granted or not. A priority encoder collects all the group requests from the single-priority arbiters and indicates among them the highest priority with its three-line output. The outputs, in addition to being forwarded to

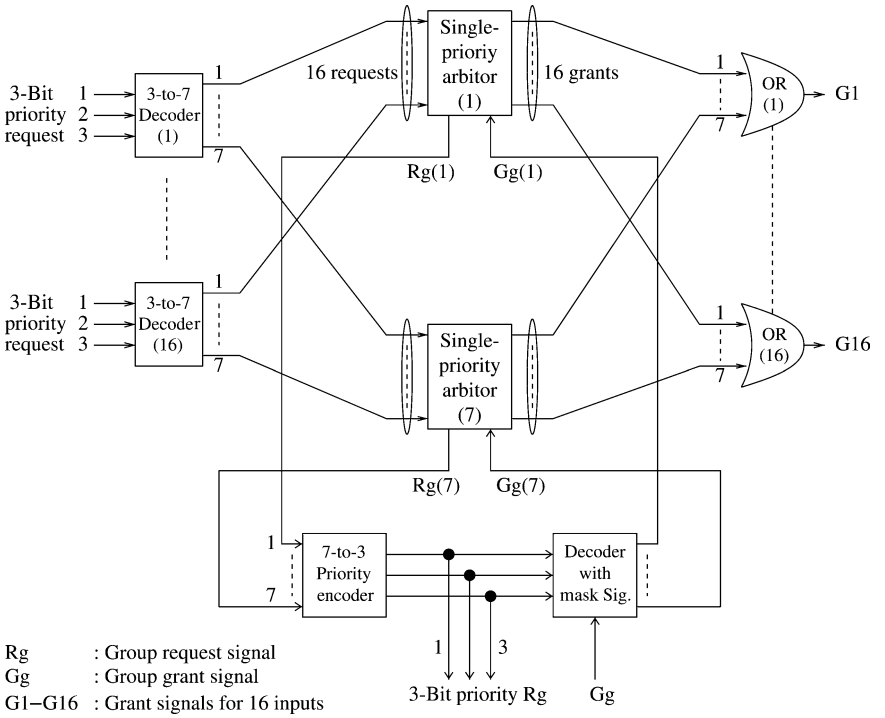


Figure 15.22 Demonstration of priority handling with parallel arbitration: seven priority levels and 16 inputs.

the upper layer, will also be used to inhibit the arbiters with lower priority from producing any active grant. A decoder with its outputs masked by the upper-layer grant signal is used to decompose the output of the priority encoder into seven single-line grant signals, each for a single-priority arbiter. Only the arbiter at the corresponding level of priority receives the upper layer's grant signal, while all the others receive nothing but a low grant signal.

15.3 OPTOELECTRONIC PACKET SWITCH CASE STUDY II

15.3.1 Petabit Photonic Packet Switch Architecture

Figure 15.23 shows the system architecture of the proposed petabit photonic packet switch, called PetaStar. The basic building modules include the input and output port controllers (IPC and OPC), input grooming and output demultiplexing modules (IGM and ODM), a photonic switch fabric (PSF), centralized packet scheduler (PS), and a system clock distribution unit. The PSF is a three-stage Clos-network with columns of input switch modules (IMs), central switch modules (CMs), and output switch modules (OMs). The PS for the three-stage Clos-network switch can be found in Chapter 12. The incoming and outgoing line rates are assumed to be 10 Gbit/s. All incoming lines will first be terminated at line cards (not shown in the figure), where packets are segmented into cells (fixed length

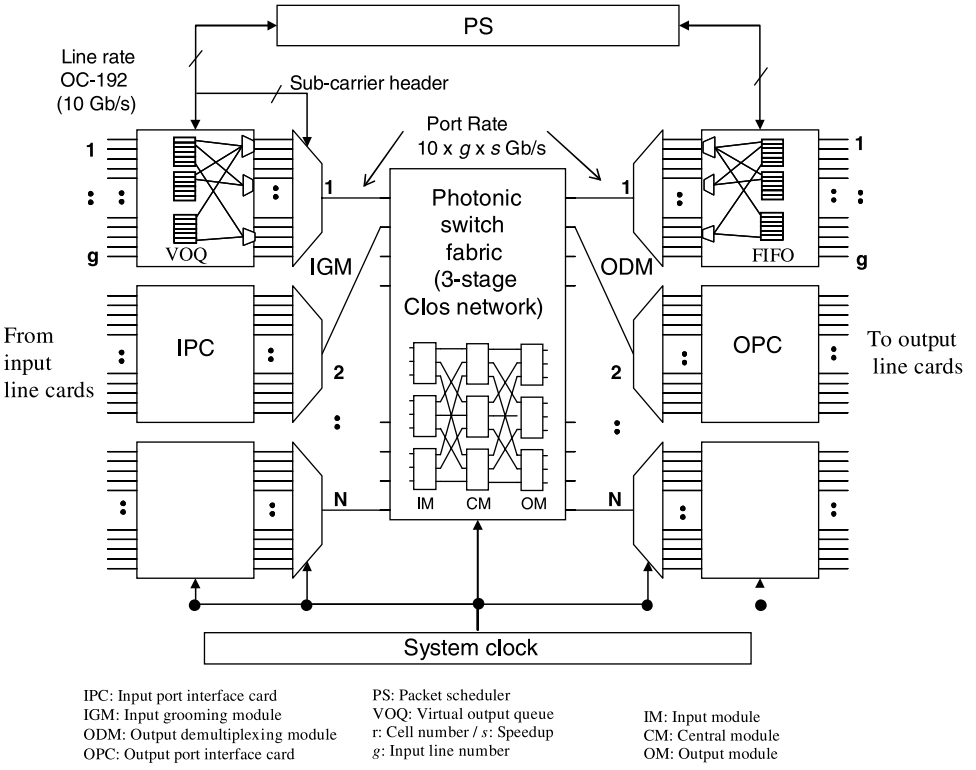


Figure 15.23 System architecture of the PetaStar.

data units) and stored in a memory. The packet headers are extracted for IP address lookup.[2] All cell buffering is implemented electronically at the IPCs and OPCs, leaving the central PSF bufferless, that is, no photonic buffering is required in the system. As a result, the bit rate of each port can operate at a speed beyond the limits of the electronics. The port speed can be equal to or greater than (with a speedup) g times the line rate, where g is the grooming factor. Virtual output queues (VOQs) at IPCs, together with the PS, provide contention resolution for the packet switch. At the input end, the majority of incoming packets are stored in the ingress line cards, where packets are segmented and stored in its VOQs. Packets destined for the same output port are stored in the same VOQ. VOQs implemented at the IPC serve as the mirror of the VOQ memory structure in the line cards. As long as they can keep the cells flowing between the line card and IPC, the size of the VOQs at the IPC can be considerably smaller than its mirror part in the ingress line cards. Buffers at the OPC are used to store cells before they are sent out to the destined egress line cards. A large buffer with a virtual input queue (VIQ) structure implemented in the line card (not shown in Fig. 15.23) is used to store the egress cells from the PSF and to re-assemble them into packets.

Figure 15.24 shows how packets flow across the system. At the input, variable-length IP packets are first segmented into cells with a fixed length of 64 bytes (including some cell

[2]Functions such as classification and traffic shaping/policing are usually performed at the edge routers, but not at core routers.

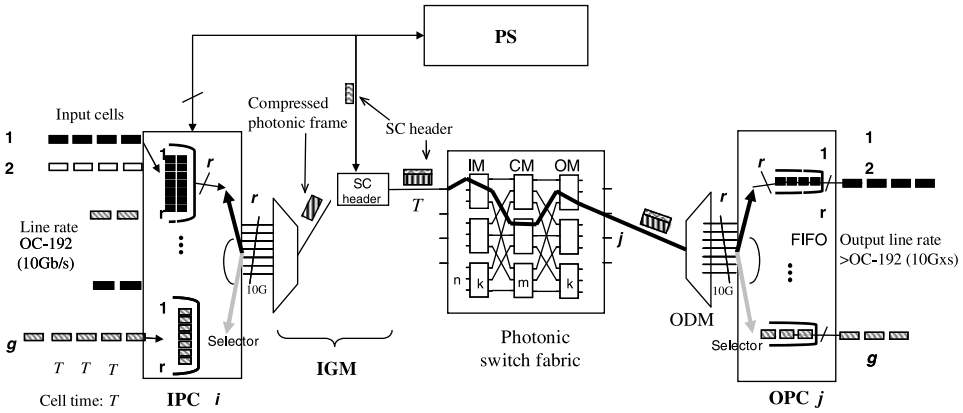


Figure 15.24 Data packet flow.

overhead), suitable to accommodate the shortest IP packet (40 bytes). At each IPC, a total of g input lines at 10 Gbit/s enter the system and terminate at the IPC. To reduce memory speed, each VOQ has a parallel memory structure to allow r cells to be read at the same time (r cells form a photonic frame). Each cell, before entering the IGM for compression, is scrambled to guarantee sufficient transitions in the data bits for optical receiver and clock recovery. In the IGM, these cells are compressed at the optical time domain to form a time-interleaved optical time division multiplex (OTDM) frame at $g \times 10$ Gbit/s. Let T be the cell time slot and $T = 51.2$ ns for 10 Gbit/s line rate. Let T_c be the compressed cell time slot at the port speed ($10 \times g \times s$ Gbit/s) and $T_c = 64\text{B}/(10 \times g \times s\,\text{Gbit/s})$. Then the compressed photonic frame period $f = r \times T_c = r \times 64\text{B}/(10 \times g \times s$ Gbit/s). With $g = r$ and $s = 1$, the frame period f is equal to the cell slot, T. Guardtime is added at the head of the frame to compensate for the phase misalignment of the photonic frames when passing through the PSF and to cover the transitions of optical devices.

At each stage of the photonic switching fabric, the corresponding sub-carrier header is extracted and processed to control the switching fabric. Since the PS has already resolved the contention, the photonic frame is able to find a path by selecting the proper output links at each stage in the switching fabric. Once the photonic frame arrives successfully at the designated output port, it is demultiplexed and converted back to r cells at 10 Gbit/s in the electronic domain by the output demultiplexing module (ODM). The OPC then forwards the cells to their corresponding line cards based on the output line numbers (OLs).

As Figures 15.23 and 15.24 show, the optical signals run between the IGM and the ODM at a rate of $g \times 10$ Gbit/s, or 160 Gbit/s for $g = 16$. All optical devices/subsystems between them operate at $g \times 10$ Gbit/s. However, the electronic devices only operate at most 10 Gbit/s (with a speedup of 1), or even lower with parallel wires, for example, four SERDES signals, each at 2.5 Gbit/s (or 3.125 Gbit/s including 8B/10B coding).

Figure 15.25 illustrates the data structure at each stage of the switch. Before the data payload, two header fields that contain the OL in the destined output port and the input line number (IL) of the switch are added to each incoming cell (see Fig. 15.25*a*). The OL is used to deliver cells to the destined output lines when the photonic frame (r cells) arrives at the OPC. A validity bit is inserted at the beginning of the cell to indicate if the cell is valid or not. The overhead bits introduced by OL and IL can be calculated as $\log_2(g)$ and $\log_2(g \times N)$, respectively. For example, for a petabit system with $N = 6400$ and $g = 16$, the cell header

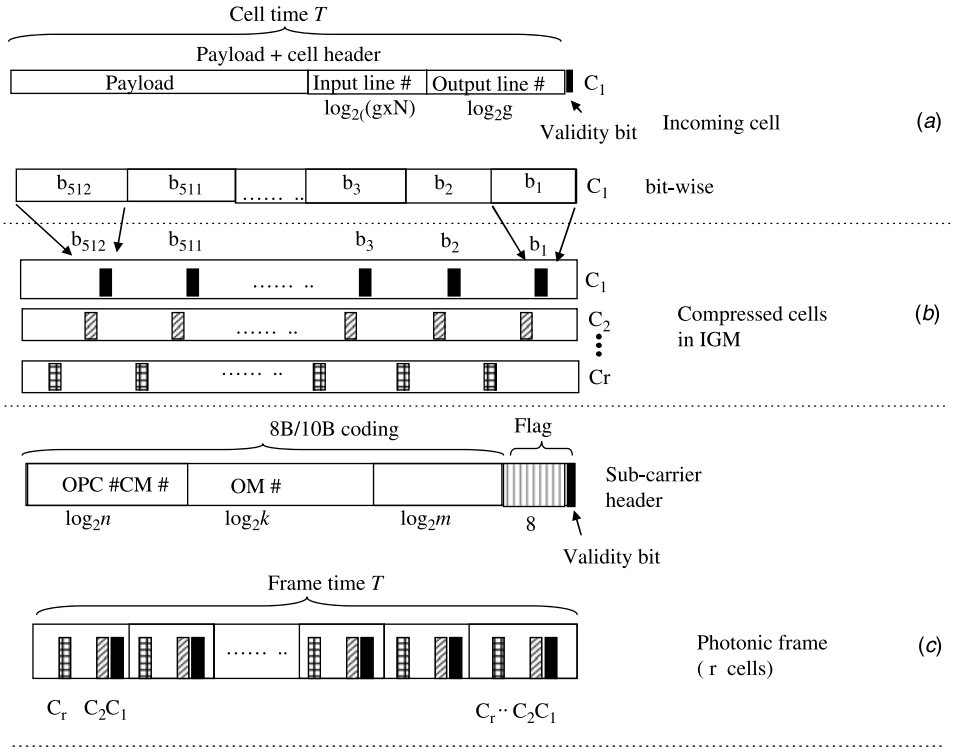


Figure 15.25 Data structure of: (*a*) Incoming cell; (*b*) Compressed cells; (*c*) Photonic frame.

length is 21 bits (1 + 4 + 16). Bits in each cell are compressed and time-interleaved using OTDM techniques in the IGM to form the photonic frames that are ready to transmit through the PSF (see Fig. 15.25*b*). Each photonic frame goes along with an out-of-band sub-carrier (SC) header. Using the photonic frame as its carrier, the SC header is amplitude-modulated on the photonic frame at a much lower sub-carrier frequency. The estimated raw bandwidth required for the SC header is about 600 MHz. Standard multi-level coding schemes can be applied to further compress the SC bandwidth to 80 MHz or less, allowing the SC header to be carried around the DC frequency. The first field in the SC header is a flag containing a specific pattern for frame delineation since the photonic frames carrying the SC header do not precisely repeat in the time domain. The payload is 8B/10B coded for correctly finding the flag. Three fields are attached to the SC header to provide routing information at each stage of the PSF. The three fields include CM, OM, and OPC numbers with $\log_2(m)$, $\log_2(k)$, and $\log_2(n)$ bits of information, where m and k are the numbers of CM and OM, and n is the number of outputs at each OM. At the beginning of the frame, a validity bit is added to indicate if the frame contains valid cells.

Figure 15.26 gives an example of how cells flow through the IPC. In this case, packets *A*, *B*, and *C* from input lines 1 and *g*, respectively, are destined for the same output port of the PSF (port 1). Packets *A* and *B* are heading towards output line 1 while packet *C* is headed towards output line *g* at the same OPC. Upon arriving at the IPC, each packet, already segmented into a number of fixed-size cells, is stored in the corresponding input line memory. In this example, packet *A* is segmented into 24 cells (cells A_0 to A_{23}), packet

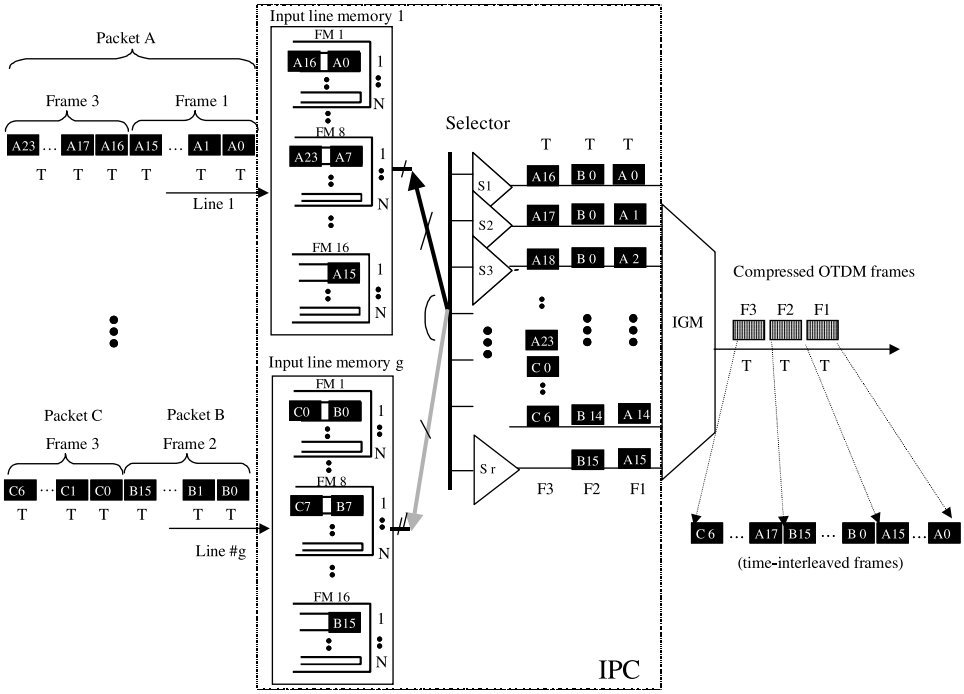


Figure 15.26 Example of illustrating how frames are formed in the PRC.

B contains 16 cells (cells B_0 to B_{15}), and packet C has eight cells (cells C_0 to C_7). All incoming cells are stored in the r frame memories in a round-robin manner.

As soon as a cell arrives at the input line memory, a request is sent to the packet scheduler that tracks all of the incoming cells. The scheduler, based on a new hierarchical frame-based exhaustive matching scheme, sends back the grant signal if the transmission has been granted. As a result, 16 cells (A_0 to A_{15}) from input line memory 1 are selected at the first frame period to form frame number 1, followed by 16 cells (B_0 to B_{15}) from input line memory g selected at the next frame period to form frame number 2. In this example, the remaining eight cells (A_{16} to A_{23}) will be aggregated with another eight cells from C packet (C_0 to C_7) to form frame number 3. The reason that packet B is granted prior to the second half of packet A is because packet B has a filled frame and thus has a higher priority for transmission. At the IGM, cells are compressed into the time-interleaved photonic frames and are thus ready to be routed through the PSF.

Following the above example, Figure 15.27 shows how packets A, B, and C are processed as they are demultiplexed at the OPC and reassembled at the egress line cards. Photonic frames containing the compressed cells are demultiplexed in the ODM and sent into r parallel inputs to the selector array. According to the cell header, A_0 to A_{15} go to the 16 FIFOs located in output line memory 1 at the first frame period. At the next frame period, B_0 to B_{15} are sent to the same 16 FIFOs in output line memory 1. At the next frame period, photonic frame number 3 arrives at the OPC. The remaining part of packet A is sent to input line memory 1, while cells from packet C go to output line memory g. These cells are then read out from the FIFOs to the designated output line (output line 1 in this case) at a speed larger than 10 Gbit/s. The VIQs at the line card are used to reassemble packets A, B, and C.

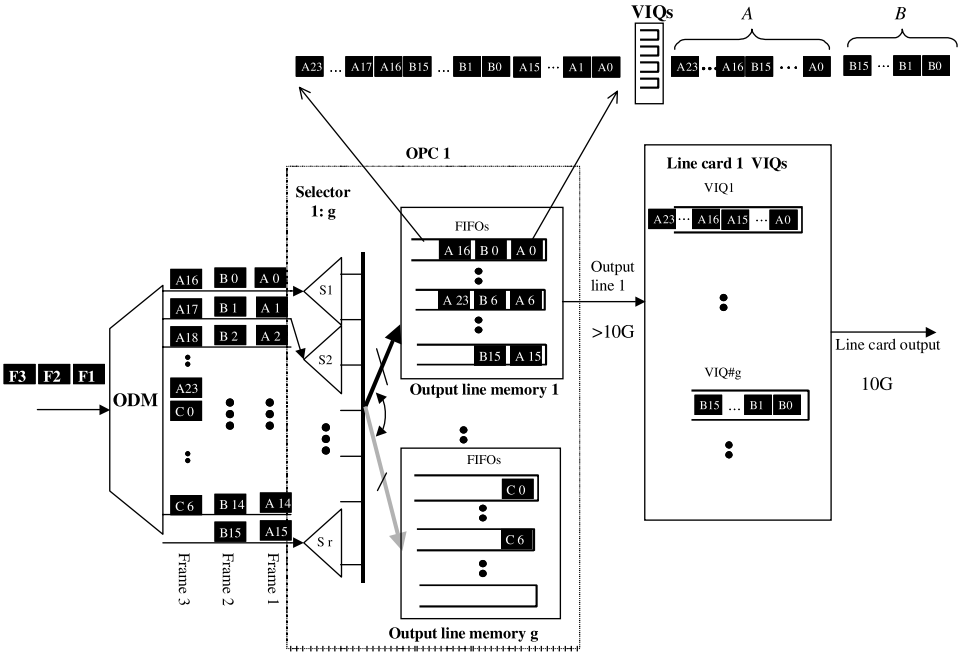


Figure 15.27 Example of illustrating how packets are demultiplexed at the OPC and resembled at the egress line cards.

Synchronization can be challenging as the system scales. To achieve synchronization, a centralized frame clock will be supplied to each module in the system. Each switching action, including buffer reading and writing, switching of laser wavelength, and OTDM multiplexing and demultiplexing, will be synchronized according to the same frame clock signal with a frequency of, for example, 1/51.2 ns, or 19.5 MHz. The clock signal will be distributed among the modules using optical signals through fibers to provide a sharp stroke edge for triggering operations. A sinusoidal signal at 10 GHz will be distributed to each module as the base frequency for the synchronization. The sub-carrier provides a trigger signal at each switching stage. Upon detecting the sub-carrier signal in the sub-carrier unit (SCU), which indicates the arrival of a photonic frame at the input of the module, the SC processor will wait for a precise time delay before starting the switching operation. The time delay through the fiber connections will be chosen precisely so that it can accommodate the longest processing delay in the header process. We will study the minimum timing tolerance contributed by both photonic and electronics devices, as well as fiber length mismatch in the system, which will ultimately determine the guardtime between the photonic frames. For instance, with a frame period of 51.2 ns and 10 percent used for the guardtime overhead, the guardtime can be 5 ns, which is sufficiently large to compensate for the phase misalignment, fiber length mismatch, and optical device transitions.

15.3.2 Photonic Switch Fabric (PSF)

Multistage PSF. Figure 15.28 shows the structure of the PSF. It consists of k IMs, m CMs, and k OMs in a three-stage Clos network. The switch dimensions of the IM and OM are

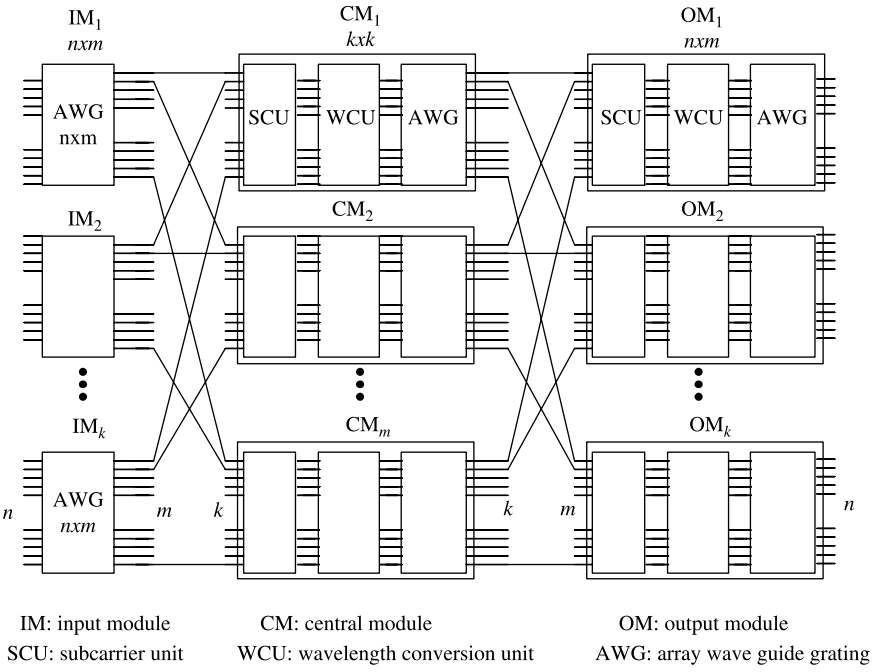


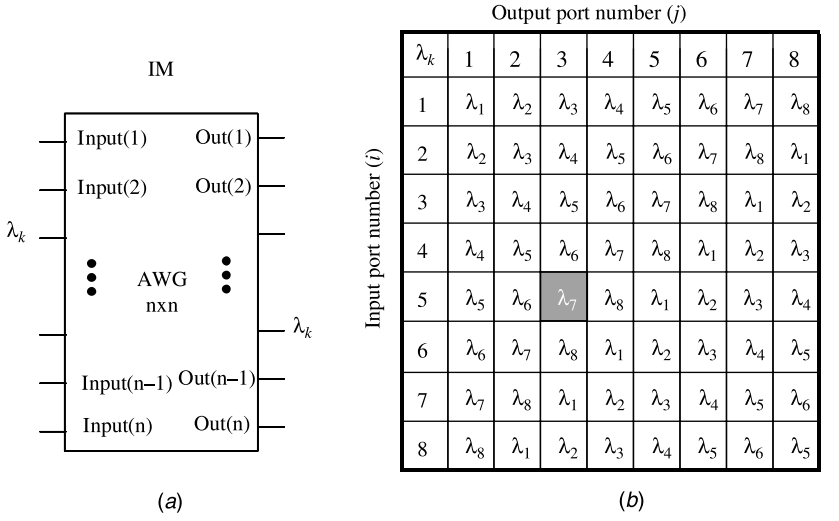
Figure 15.28 PSF architecture.

$n \times m$ while CM is $k \times k$. The IM at the first stage is a simple AWG device. The CMs and OMs consist of a SCU, a wavelength switching unit (WSU), and an AWG. A 6400×6400 switch can be realized using 80 wavelengths, that is, $n = m = k = 80$. With the port speed of 160 Gbit/s, total switch capacity reaches 1.024 petabit/s.

Based on the cyclic routing property of an $n \times n$ AWG router, full connectivity between the inputs and outputs of the IM can be established by arranging input wavelengths. By switching the laser wavelength at each of the n inputs, the incoming optical signal that carries the photonic frame can emerge at any one of the n outputs, resulting in an $n \times n$ non-blocking space switch. Since the AWG is a passive device, the reconfiguration of this space switch is solely determined by the active wavelength tuning of the input tunable laser. The wavelength switching can be reduced to a couple of nanoseconds by rapidly changing the control currents for multiple sessions in tunable semiconductor lasers [21–23].

An example of a wavelength routing table for an 8×8 AWG is shown in Figure 15.29. A wavelength routing table can be established to map the inputs and outputs on a specific wavelength plan. In general, the wavelength λ_k from input $i\,(i = 1, 2, \ldots, n)$ to output $j\,(j = 1, 2, \ldots, n)$ can be calculated according to the following formula: $k = (i + j - 1)$ modulo n. For example, input (5) needs to switch to wavelength λ_7 to connect to output (3) for $n = 8$ as highlighted in the router table.

To cascade AWGs for multi-stage switching, CMs and OMs have to add wavelength conversion capability, where the incoming wavelengths from the previous stage are converted to new wavelengths. An all-optical technique is deployed to provide the necessary wavelength conversion without O/E conversion. Figure 15.30 illustrates the detailed design of the CM and OM. Three key elements used to implement the switch module are the SCU for header processing and recognition, the wavelength conversion unit (WCU) for performing



λ_k	1	2	3	4	5	6	7	8
1	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8
2	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	λ_1
3	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	λ_1	λ_2
4	λ_4	λ_5	λ_6	λ_7	λ_8	λ_1	λ_2	λ_3
5	λ_5	λ_6	λ_7	λ_8	λ_1	λ_2	λ_3	λ_4
6	λ_6	λ_7	λ_8	λ_1	λ_2	λ_3	λ_4	λ_5
7	λ_7	λ_8	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
8	λ_8	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_5

Figure 15.29 Each input can use the wavelength assigned in the table to switch to any one of eight outputs. (*a*) IM based on an $n \times n$ AWG router; (*b*) Example of 8×8 wavelength routing table.

all-optical wavelength conversion, and the AWG as a space switch (the same as the one in the IM). The main function of the SCU is to process the SC header information for setting-up the switch path. The SC header information, which consists of 3 bytes, is readily available at each stage of the PSF as these bytes are carried out-of-band along with each photonic

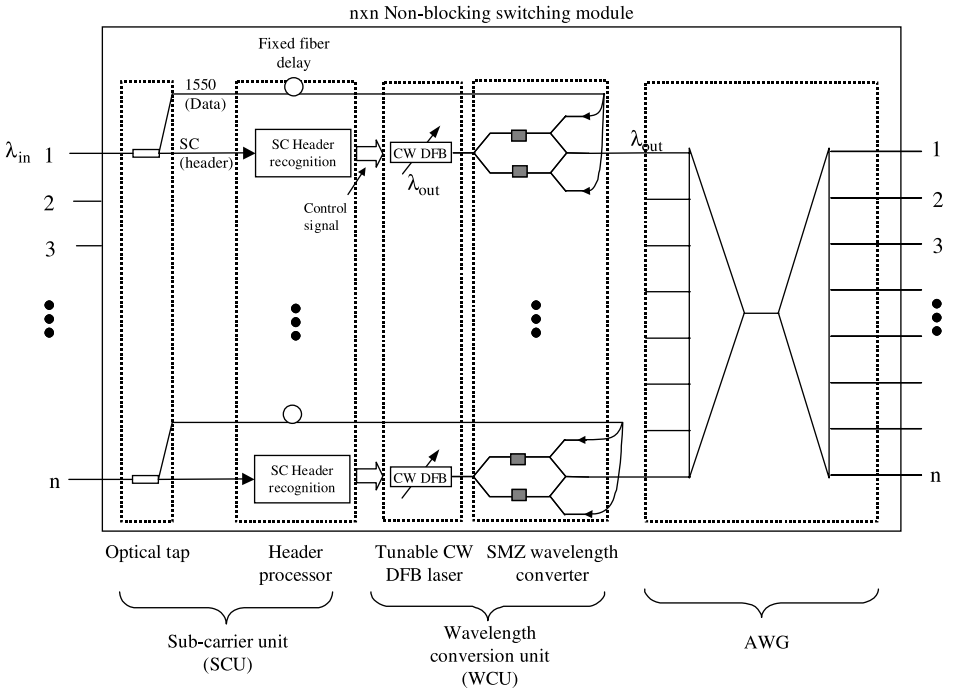


Figure 15.30 Structure of CM and OM.

frame as shown in Figure 15.25*c*. Upon arriving at each module, a portion of the power from the photonic frame is stripped by an optical tap and fed into the SCU for sub-carrier demodulation. At the front end of the SCU, a low-bandwidth photo-detector and low-pass filter are able to recover the header information from the photonic frame. The SC header information is used to set the wavelength of the continuous-wave (CW) tunable laser in the WSU. On the data path, a fixed fiber delay is added to allow the SCU to have sufficient time to perform header recognition and processing. The total propagation time between the input and output links is properly controlled to guarantee that the frame arrives at each switch module within system timing tolerance.

Recently, wavelength conversion at the OTDM rate up to 168 Gbit/s was demonstrated that used a symmetric Mach-Zehnder (SMZ)-type all-optical switch [24, 25]. The strong refractive index change from the carrier-induced resonance nonlinearity in the SOAs, coupled with the differential interferometric effect, provides an excellent platform for high-speed signal processing. A similar device has also been demonstrated in demultiplexing an ultra-high bit rate OTDM signal at 250 Gbit/s, which shows its excellent high-speed capability. Therefore, we can consider using an array of such devices to accomplish the all-optical wavelength conversion at an ultra-high bit rate.

The basic structure of the WCU, based on a Mech-Zehnder (MZ) interferometer with in-line SOAs at each arm, is shown in Figure 15.31. The incoming signal with wavelength (λ_{old}) is split and injected into the signal inputs, entering the MZ interferometer from the opposite side of the switch. Figure 15.31*b* shows the operation of the wavelength conversion. A switching window at time domain can be set up (rising edge) by the femto-second ultrafast response induced by the signal pulses through the carrier resonance effect of SOAs. The fast response of the SOA resonance is in the femto-second regime, considerably shorter than the desired rise time of the switching window. Although the resonance effect of each individual SOA suffers from a slow tailing response (100 picoseconds), the delayed differential phase in the MZ interferometer is able to cancel the slow-trailing effects, resulting in a fast response on the trailing edge of the switching window. By controlling the differential time between the two SOAs accurately, the falling edge of the switching window can be set at the picosecond time scale. The timing offset between two SOAs located at each arm of the MZ interferometer controls the width of the switching window. To be able to precisely control the differential timing between two arms, a phase shifter is also integrated in the interferometer. The wavelength conversion occurs when a CW light at a new wavelength (λ_{new}) enters the input of the MZ interferometer. An ultrafast data stream whose pattern is the exact copy of the signal pulses at λ_{old} is created with the new wavelength at the output of the MZ interferometer (marked switched output in the figure), completing the wavelength conversion from λ_{old} to λ_{new}.

Using active elements (SOAs) in the WCU greatly increases the power budget while minimizing the possible coherent crosstalk in the multi-stage PSF. As shown in Figure 15.31*a*, the incoming signal pulses, counter-propagating with the CW light from the tunable DFB laser, eventually emerge at the opposite side of the WCU, eliminating the crosstalk between the incoming and the converted outgoing wavelengths. The required switching energy from the incoming signal pulses can be as low as a couple of femto-joules due to the large resonance non-linearity. After the wavelength conversion, the output power level for the new wavelength may reach mW-level coming from the CW laser. Therefore, an effective gain of 15 to 25 dB can be expected between the input and output optical signals through the WCU. This effective amplification is the key to the massive interconnected PSF maintaining effective power levels for the optical processing at each stage. The building modules

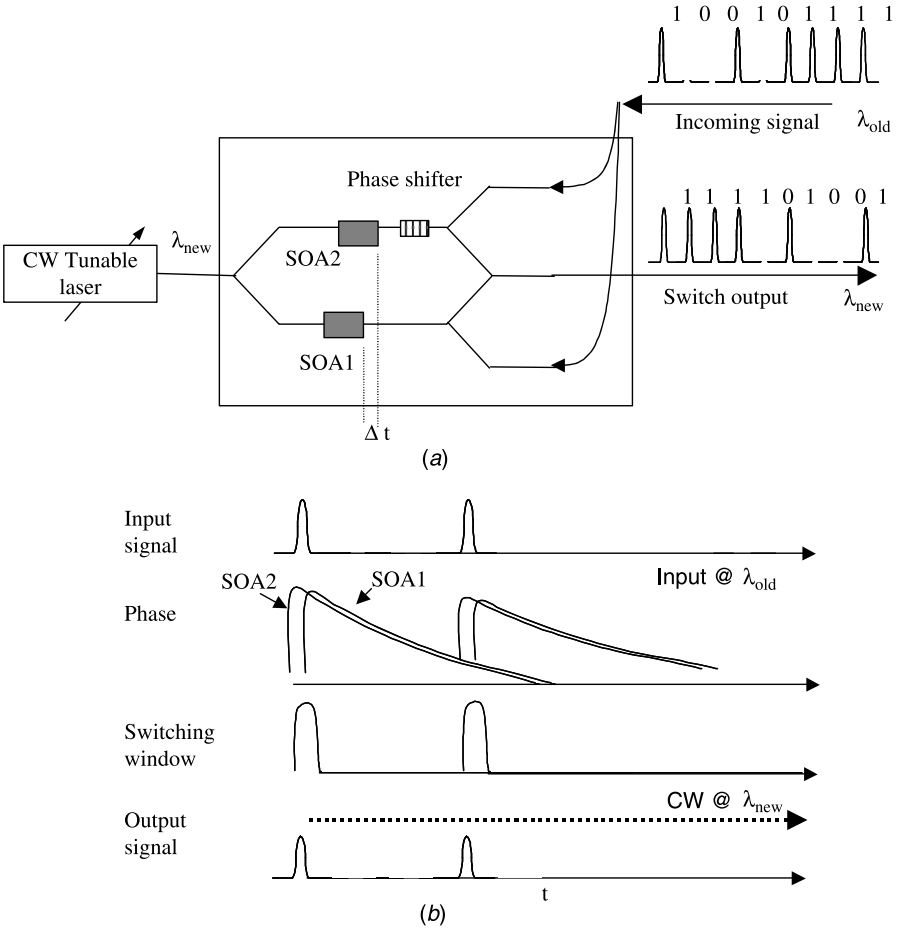


Figure 15.31 (*a*) Wavelength conversion unit based on an SOA Mach-Zehnder interferometer with differential time delay between two arms. The input signal is λ_{old} while the output signal is converted to a new wavelength (λ_{new}) determined by the CW tunable laser at the device input. (*b*) Timing diagram of the ultrafast wavelength conversion process.

used in the PSF have the potential to be monolithically integrated due to their similar architectures. There have already been attempts to build integrated SOAs in a waveguide structure on PLC technologies [26]. As shown in Figure 15.30, the components in the dashed lines are the best candidates for integration due to their similarity in architecture and design. This integration provides dramatic savings on the power budget and component cost.

To reach a total switch capacity of 1.024 petabit/s, the required bandwidth can be estimated to be 192 nm assuming 80 wavelengths at a 160-Gbit/s port speed, ultimately limiting the system scalability. It is necessary to apply techniques such as polarization multiplexing and the binary coding scheme to further reduce the total spectral width by a factor of two or more. The tuning range of laser and SOA also limits the scalability. However, we propose to use multiple components for tunable laser and SOAs, each of which is capable of tuning over a subset wavelength of the whole spectrum.

OTDM Input Grooming Module (IGM). Optical time division multiplexing (OTDM) can operate at ultrafast bit rates that are beyond the current electronics limit, which is around 40 Gbit/s. By interleaving short optical pulses at the time domain, aggregated frames can be formed to carry data at bit rates of hundreds of gigabits per second. Using the OTDM technique, there can be at least one order of magnitude in bandwidth increase compared with the existing electronics approach.

The IGM interfaces with r parallel electronic inputs from the IPC. Figure 15.32 shows the structure of the IGM based on the OTDM technology. It consists of a short-pulse generation unit, modulator array, and a passive $r \times r$ fiber coupler with proper time delays for time-interleaved multiplexing. Optical pulses with widths of several picoseconds can be generated using electro-absorption modulators (EAMs) over-driven by a 10-GHz sinusoidal clock signal. Using a tunable CW DFB laser as the light source, the wavelength of the output ultra-short pulses can also be tunable. The pulse width will be around 7–10 picoseconds generated by the cascaded EAMs, which is suitable for data rates up to 100 Gbit/s. To generate pulses suitable for higher bit rates (>100 Gbit/s), nonlinear compression with self-phase modulation (SPM) can be used. The pulses, generated from the EAMs, are injected into a nonlinear medium (a dispersion shifted or photonic bandgap fiber) followed by a compression fiber (dispersion compensation fiber) to further compress the pulse width to about 1 picosecond. The parallel r input lines from the IPC electronically modulates the modulator array to encode the bit stream onto the optical pulse train. Precise time delays on each branch of the fiber coupler ensure time-division multiplexing of r inputs. Through the parallel-to-serial conversion in the multiplexer, r cells at 10 Gbit/s from the IPC are now effectively compressed in the time domain as the RZ-type photonic frame that operates at

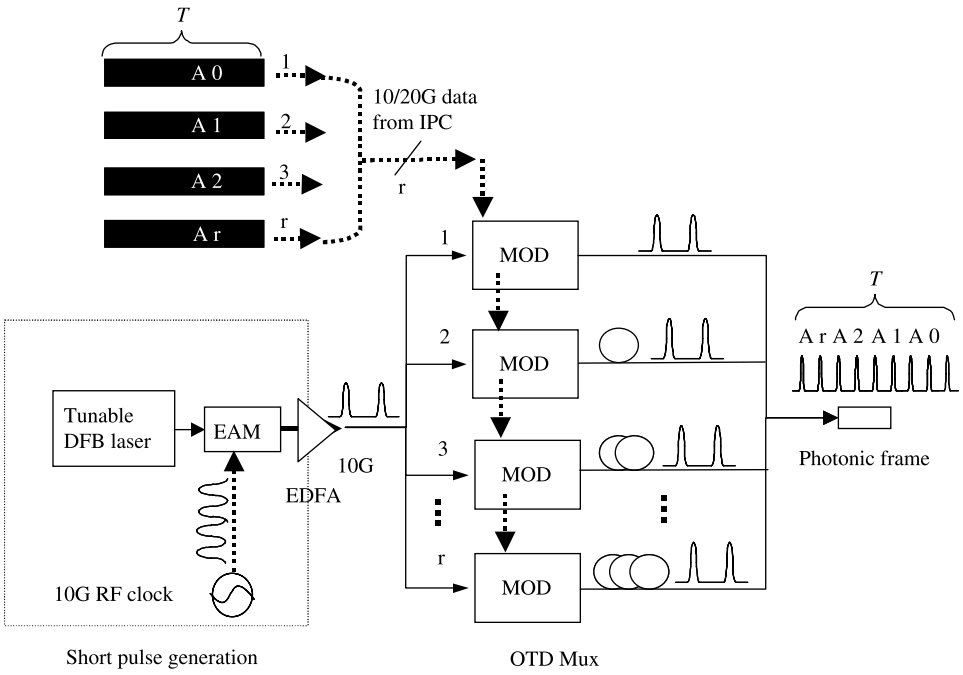


Figure 15.32 Structure of the IGM based on optical time-division multiplexing.

$r \times 10$ Gbit/s in serial. The fiber coupler and time delays can be integrated using planar waveguide structures [26].

OTDM Output Demultiplexing Module (ODM). At the receiving end of the system, the ODM demultiplexes photonic frames from the output of the PSF into r parallel electronic signals at 10 Gbit/s. As Figure 15.33 shows, the ODM consists of a quarter-phase detector and quarter-phase shifter, an array of OTDM demultiplexers (DEMUX) based on EAMs, and the photo-detector (PD) array for O/E conversions. We have previously demonstrated ultrafast demultiplexing at 40, 80, 100, and 160 Gbit/s using cascaded EAMs as the gating device. As shown in the inset of Figure 15.33, the OTDM demultiplexer consists of two cascaded EAMs based on multiple quantum well devices [27, 28]. An SOA section is also integrated with the EAM to provide optical amplification at each stage. The optical transmission of the EAM, controlled by the driving electronic signal, responds highly nonlinearly and produces an ultra-short gating window in the time domain. Cascading the EAMs can further shorten the gating window compared to a single EAM. The incoming optical signal is split by a $1 \times r$ optical coupler into r modulators located in the array structure. Each EAM is over-driven by a 10-GHz sinusoidal radiofrequency (RF) clock to create the gating window for performing demultiplexing. The RF driving signals supplied to adjacent modulators in the array structure are shifted by a time τ, where τ is the bit interval inside the photonic frame. As a result, r modulators are able to perform demultiplexing from $r \times 10$ Gbit/s down to 10 Gbit/s on consecutive time slots of the photonic frame.

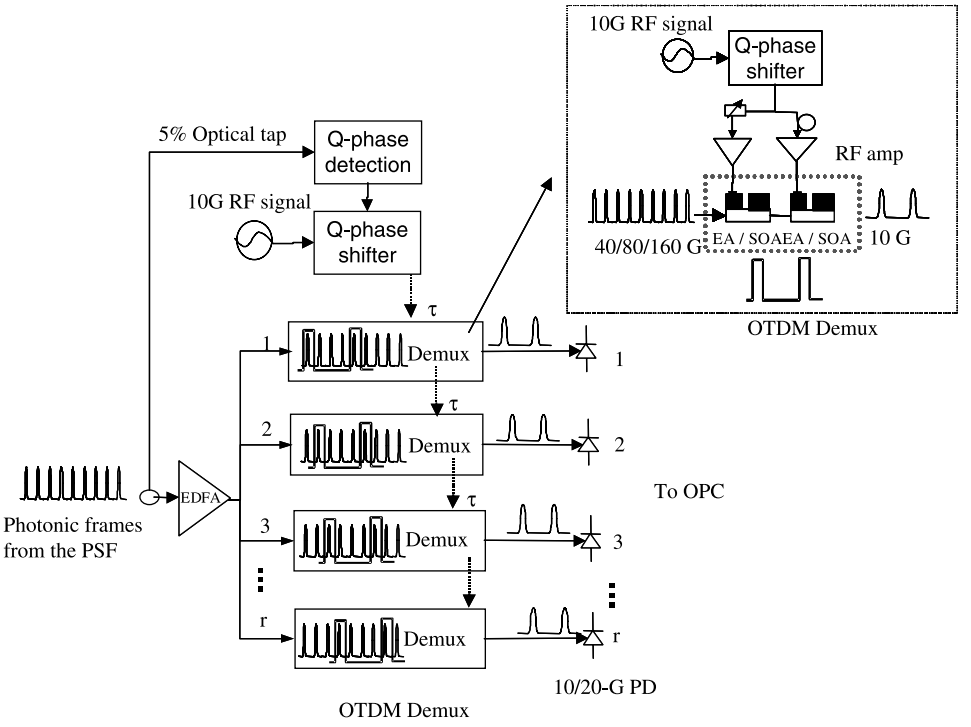


Figure 15.33 Structure of the ODM. The OTDM demultiplexer, based on cascaded EAMs, is shown in the inset.

The incoming frames may inherit timing jitters induced by either slow thermal effects (fiber, device, and component thermal lengthening) or system timing errors. The result is a slow (compared to the bit rate) walk-off from the initial timing (phase). Since the frames are operating on a burst mode, traditional phase lock loop cannot be applied here. To track the slow varying jitters on the burst frames, we suggest a quarter-phase locking scheme using phase detection and a shifter.

A quarter-phase detector is shown in Figure 15.34. Four OTDM demultiplexers, based on EAM technology, are used as the phase detectors because of their high-speed gating capability. The driving RF sinusoidal signal for each modulator is now shifted by $(\frac{1}{4})\tau$. Depending on the phase (timing) of the incoming signal, one of the four demultiplexer outputs has the strongest signal intensity compared to the three other detectors. A 4 : 2-bit decoder is then used to control the quarter-phase shifter to align the 10-GHz RF signal to the chosen phase. For example, assuming EA_1 aligns best with the incoming signal at one incident, output from Q_0 would be the strongest signal and would be picked up by the comparator. The clock that is supplied to the OTDM demux is then adjusted according to the detected phase.

The quarter-phase shifter, also shown in Figure 15.34, is used to rapidly shift the phase according to the detected phase. The quarter-phase shifter has been demonstrated using a digital RF switched delay lattice. The semiconductor switch is used to set the state at each stage. Depending on the total delay through the lattice, the output phase can be shifted by

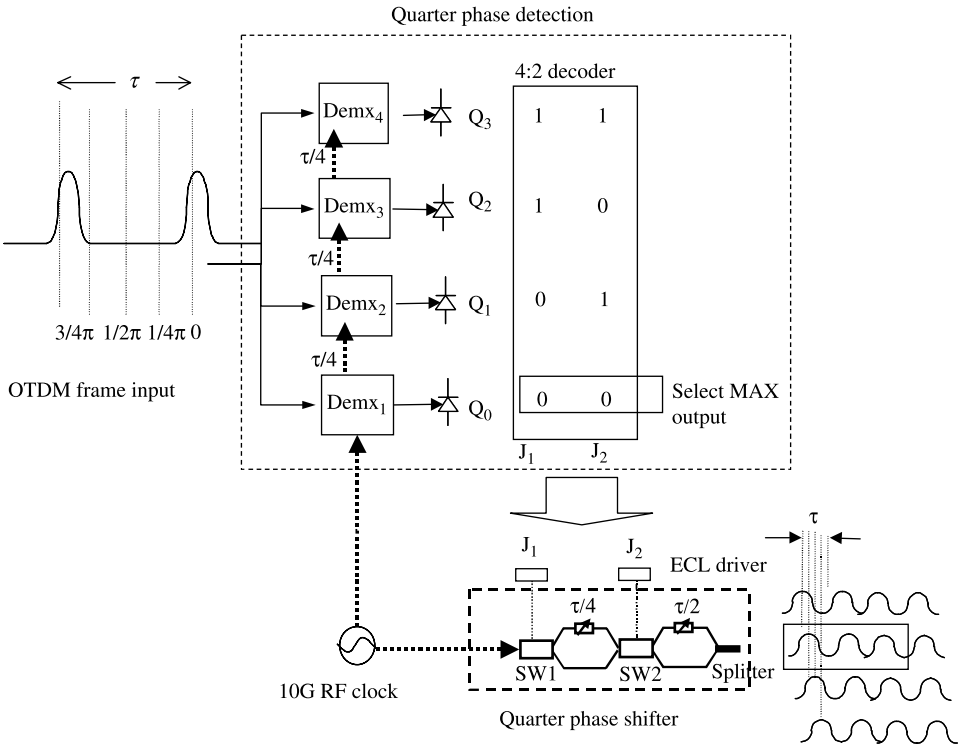


Figure 15.34 Quarter-phase detection and rapid RF phase shifter. SW = semiconductor switch.

changing the state at each lattice. The resulting clock is synchronized with the incoming packet with a timing error less than $\pm 1/8\tau$.

15.4 ALL OPTICAL PACKET SWITCHES

In optical packet switches, logical control and contention resolution are handled by an electronic controller and packets are carried and stored in optical memories. There are two kinds of optical memory used in the all optical packet switches; one is the traveling type based on fiber delay lines and the other is based on the fiber-loop type where packets, carried in different wavelengths, co-exist in the fiber loop.

15.4.1 The Staggering Switch

The staggering switch [29] is one of the optically transparent switches. The major components of the switch are splitter detectors, rearrangeable nonblocking switches, and a control unit. The switch architecture is based on two stages: the scheduling stage and the switching stage, as shown in Figure 15.35. These two stages could be considered as rearrangeably nonblocking networks. The scheduling stage and the switching stage are of size $N \times M$ and $M \times N$, respectively, where M is less than N. These two stages are connected by a set of optical delay lines having unequal delay. The idea behind this architecture is to arrange incoming cells in the scheduling stage in such a way that there will be no output-port collision in the switching stage. This is achieved by holding the cells that cause output port collision on the delay lines. The delay on the delay line d_i is equal to i cell slots. The arrangement of incoming cells is accomplished electronically by the control unit according to the output port requests of incoming cells.

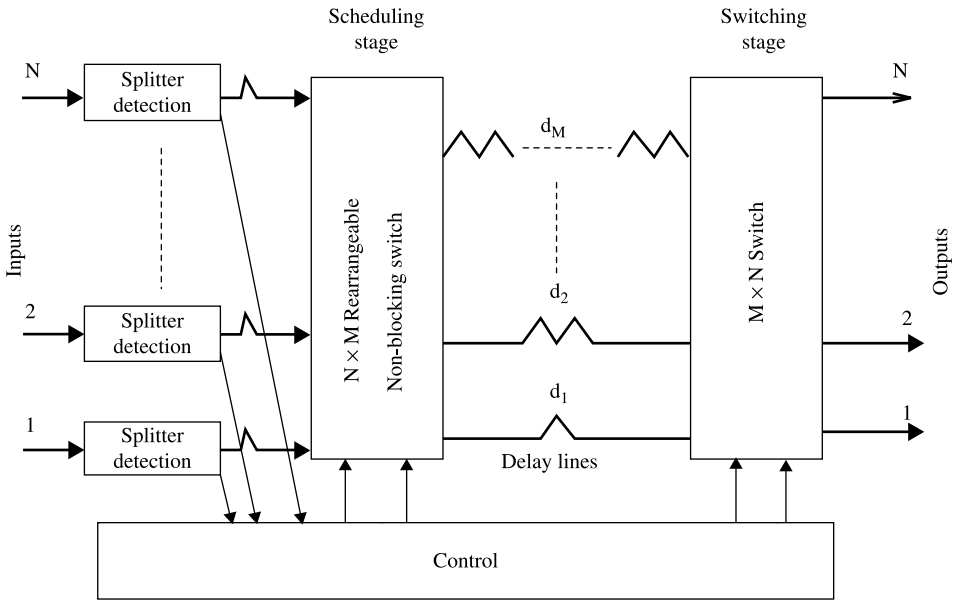


Figure 15.35 Block diagram of the staggering switch (© 1993 IEEE).

When a cell arrives at the switch, its header information is converted into electrical signal and sent to the control unit by the corresponding splitter detector. After evaluating the current destination requests considering the previous requests, the control unit sends the information related to the current schedule to the scheduling stage. The cell is routed through the scheduling stage with respect to the information sent by the control unit. Due to the statistical properties of the incoming cells, it is possible to lose some cells in the scheduling stage. After waiting for a certain period of time on the assigned delay line, the cell reaches the switching stage. No contention occurs in the switching stage due to the precautions taken by the control unit, and the cell reaches the requested output port. In this architecture, cells arriving at the same input port may arrive at output ports in the reverse order since they are assigned to different delay lines. Ordered delivery of cells at the output ports can be achieved by some additional operations in the control unit.

The main bottleneck in this switch architecture is the control unit. The proposed collision resolution algorithm is too complicated to handle large switch size or high input line rate. Some input buffers may be necessary in order to keep newly arriving cells while the control unit makes its arrangements.

15.4.2 ATMOS

Chiaroni et al. [30] proposed a 16 × 16 photonic ATM switching architecture for bit rates up to 10 Gbit/s. Basically, this switch consists of three main blocks: (1) the wavelength encoding block, (2) the buffering and time switching block, and (3) the wavelength selection block as shown in Figure 15.36. In the wavelength encoding block, there are N wavelength converters – one per input. Each input is assigned a fixed wavelength by its wavelength converter. When a cell arrives, a small power of optical signal is tapped by a coupler and converted to electronic signal. There is a controller that processes this converted data and extracts the routing information for the cell. The arriving cells with different wavelengths are

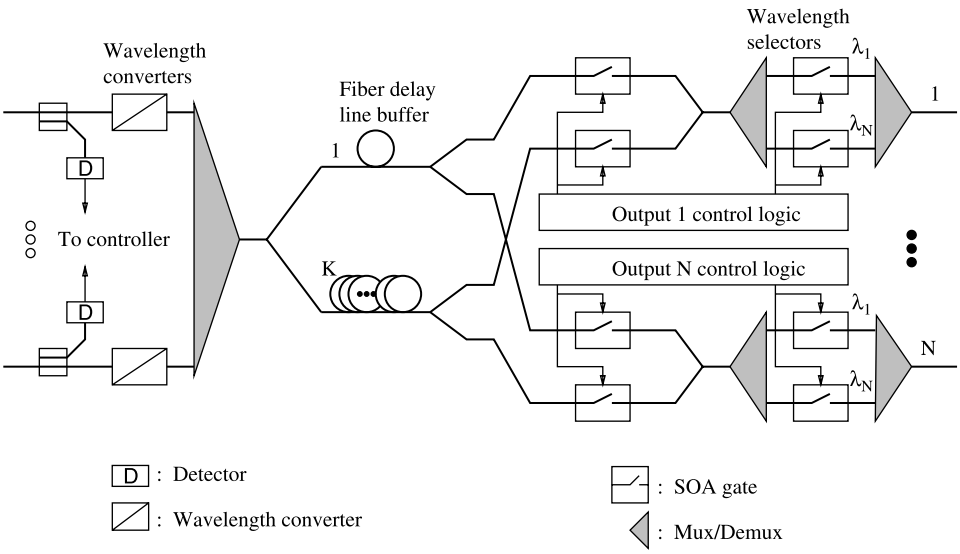


Figure 15.36 Architecture of the ATMOS switch.

wavelength-division multiplexed in the buffering and switching block by a multiplexer. The buffering and time switching block contains K fiber delay lines to store the payloads of the incoming cells. There is also a space switch that is made of SOA gates. These gates are used to select the cells from the fiber delay lines and route them to the requested output ports. The wavelength selection block consists of multiplexer/demultiplexer and SOA gates in order to select a specific wavelength destined for an output port in a cell time slot. This switch can perform the multicast function by using the broadcast and select approach.

The cell contention problem is solved by the fiber delay lines. However, as this fiber delay line approach cannot provide the sharing property, a great number of delay lines are necessary to meet the cell loss requirement. The architecture is bulky in structure and the switch size is limited by the available number of wavelengths.

15.4.3 Duan's Switch

Duan et al. [31] introduced a 16×16 photonic ATM switching architecture, as shown in Figure 15.37, where each output port is assigned a fixed wavelength. This switch consists of three main blocks: wavelength encoding block, spatial switch block, and wavelength selection block. In the wavelength encoding block, there are N wavelength converters, one per input and each being tuned to the destined output port. When a cell arrives, a small part of the wavelength is tapped by a coupler and sent to the electronic control unit, which processes the routing information of the cell. In a specific cell slot time, cells destined for different outputs are tuned to different wavelengths. These cells with different wavelengths are routed through the shortest path, which is selected by the SOA gates in the spatial switch. The spatial switch block contains K fiber delay lines to store the payloads of the cells for contention resolution. Each fiber delay line can store up to N different wavelengths. In the wavelength selection block, in each cell slot time, multiple wavelengths are broadcast to all output ports by a star coupler. There is a fixed wavelength filter at each output port. These

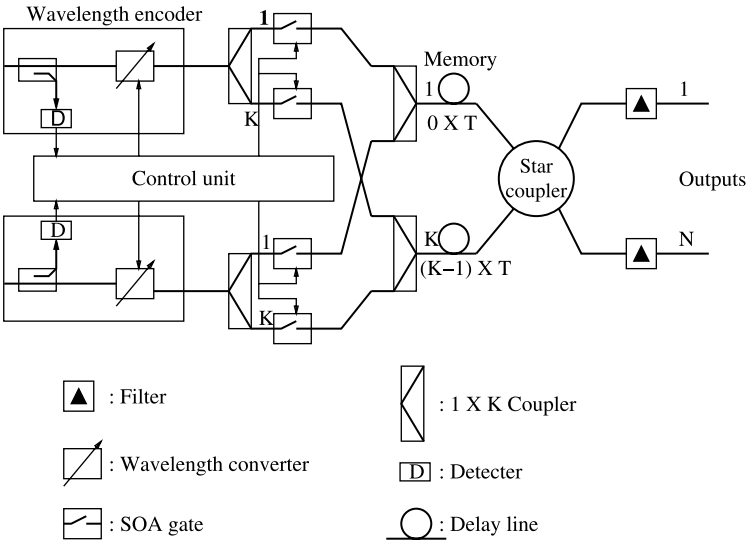


Figure 15.37 Architecture of the ATM wavelength routing system.

wavelength filters select the cells associated with their wavelengths and send them to the corresponding output ports.

This switch cannot perform multicast functions because of the fixed wavelength filters at the output ports. Furthermore, if there is more than one cell destined to the same output port, an arbitration mechanism is necessary in order to assign the incoming cells with the same wavelength to different fiber delay lines. Such a requirement increases the control complexity. In order to meet the cell loss requirement, more fiber delay lines are necessary. Moreover, the electronic controller always has to monitor the status of fiber delay lines to preserve the cell sequence.

15.4.4 3M Switch

Figure 15.38 shows the architecture of an enhanced $N \times N$ 3M switch, where incoming cells running at 2.5 Gbit/s are optically split into two paths. Cells on the top path remain in the optical domain and are routed through the optical switch plane. Cells on the bottom path are converted to the electronic domain, where their headers are extracted for processing (e.g., finding the output ports for which the cells are destined and finding new virtual path identifier/virtual channel identifier (VPI/VCI) values to replace the old VPI/VCI values). An electronic central controller, as shown in Figure 15.38, performs cell delineation, VCI-overwrite, cell synchronization, and routing. The first three functions are implemented in the photonic ATM front-end processor, while the last one is handled by a route controller that routes cells to proper output ports.

As shown in Figure 15.39, the cell format adopted in the system has 64 bytes with 5 bytes of header, 48 bytes of payload, and two guard time fields (with all ones), which are 6 and

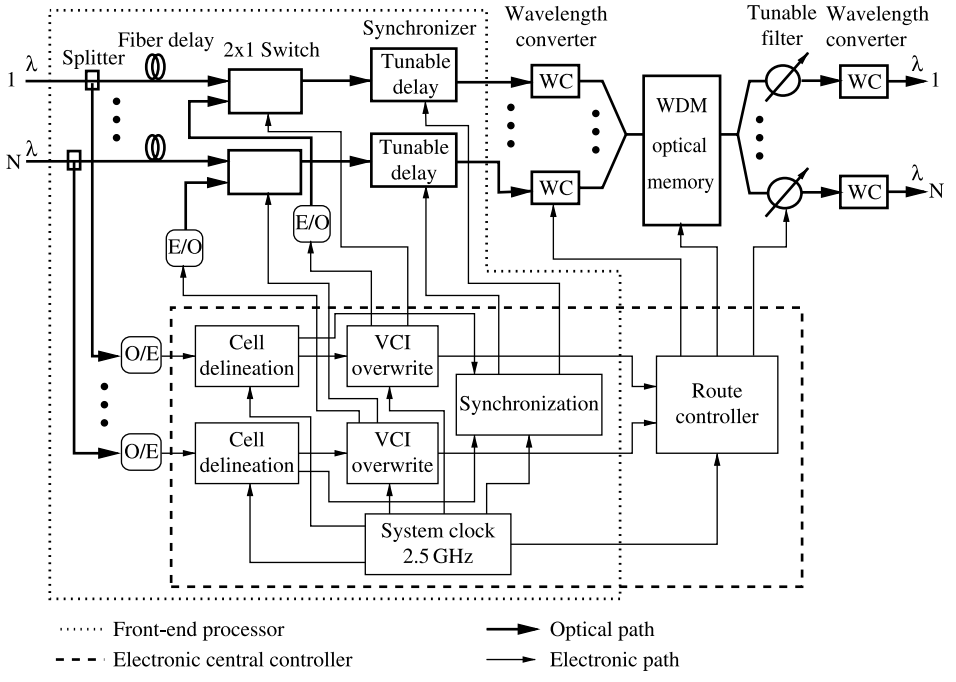


Figure 15.38 Architecture of the WDM ATM multicast (3M) switch (© 2000 IEEE).

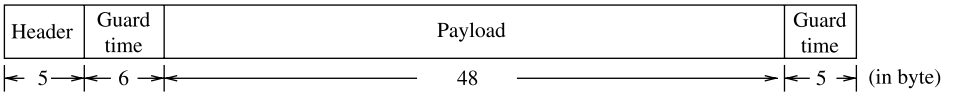
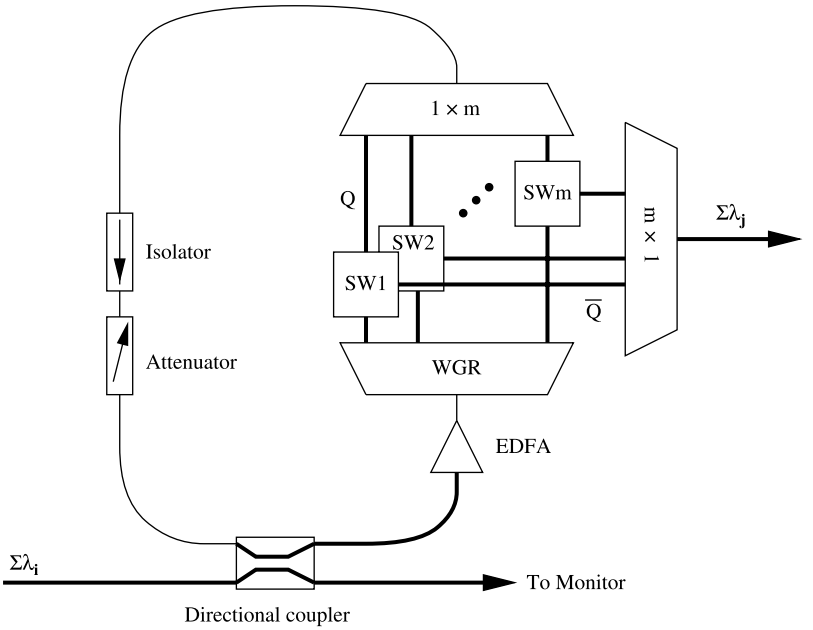


Figure 15.39 Cell format adopted in the system.

5 bytes long, respectively. The guard times are used to accommodate the slow switching of optical devices, such as optical tunable filters. The lengths of the guard times between the cells, and between the cell header and the payload were arbitrarily chosen. Cells are transmitted back-to-back and not carried in synchronous optical network (SONET) frames. Not using SONET frames eliminates the possibility of having variable gaps between or within cells caused by the need to carry SONET transport and path overhead ranging from 1 to 49 bytes.

The incoming optical cells are first delayed by fiber lines, processed for their headers, and synchronized in the front-end processor before they are sent to the switch fabric. In the switch fabric, cells are converted to different wavelengths by wavelength converters (WCs) that are controlled by the route controller, which keeps track of the available wavelengths in the WDM optical shared memory. It is a fiber loop memory, as shown in Figure 15.40, and is used to store optical cells until they are ready to be transmitted to the next node. Using a 3-dB directional coupler, cells are coupled into the optical memory and co-exist with the existing cells. Accessing cells in the optical memory is done by controlling the 1×2 space



SW : Space switch (e.g., 1×2 SOA gate)
WGR : Waveguide grating router
EDFA : Erbium doped fiber amplifier

Figure 15.40 Optical random access memory (© 2000 IEEE).

switches (SWs), for example, a SOA gate. The wavelengh-division multiplexed cell stream is amplified by an EDFA to compensate for power loss when looping in the memory. The cell stream is then demultiplexed by a waveguide grating router (WGR) into m different channels, each carrying one cell. The maximum number of cells (i.e., wavelengths) simultaneously stored in this memory has been demonstrated to be 23 circulations at 2.5 Gbit/s. Cells read from the WDM optical shared memory are broadcast to all N output ports by a $1 \times N$ splitter and selected by the destined output port (or ports, if multicast) through tunable filters that are tuned by the route controller on a per-cell basis. The final wavelength converter stage converts cells to their predetermined wavelengths. Other optical loop memory can be found in Refs. [32–36].

Figure 15.41 shows how the shared memory is controlled by a route controller. R_1–R_4 signals carry the output port addresses for which the cells are destined. An idle wavelength FIFO keeps track of available wavelengths in the memory. When up to four incoming cells arrive, free wavelengths are provided by the idle wavelength FIFO, and are used to convert incoming cells' wavelengths so they can be written to the loop memory at the same time. These wavelengths are stored in the FIFOs (FIFO 1–FIFO 4) according to the R_1–R_4 values. Since the 3M switch supports multicasting, the same wavelength can be written into multiple FIFOs. All the FIFOs (including the idle wavelength FIFO) have the same depth, storing up to m wavelengths. While the wavelength values are written sequentially (up to four writes in each cell slot) to the FIFOs, the wavelengths of the HOL cells of the FIFOs are read simultaneously so that up to four cells can be read out simultaneously. They are, in turn, used to control the tunable filters to direct the cells to the proper output ports. The write controller and read controller generate proper signals to coordinate all functional blocks.

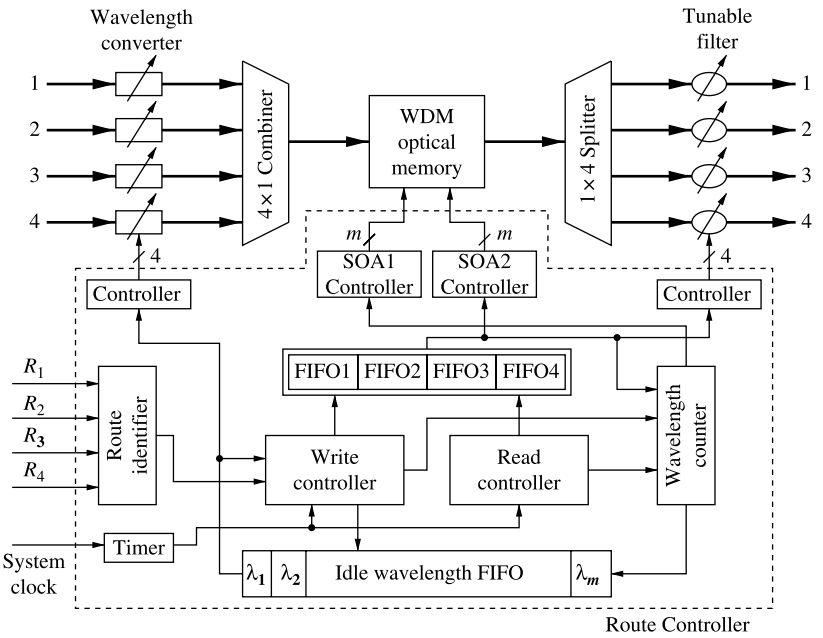


Figure 15.41 Optical shared memory controlled by a route controller (© 2000 IEEE).

15.5 OPTICAL PACKET SWITCH WITH SHARED FIBER DELAY LINES SINGLE-STAGE CASE

15.5.1 Optical Cell Switch Architecture

To buffer cells, optical timeslot interchangers (OTSIs) have been widely employed [37]. An OTSI is a single-input-single-output optical (SISO) device that consists of a number of fibre delay lines (FDLs). Let T_{cell} be the duration of each timeslot and $(F-1)T_{\text{cell}}$ the maximum delay that can be imposed on a cell. Figures 15.42 and 15.43 depict a nonblocking OTSI and a blocking OTSI, respectively. An OTSI is said to be nonblocking if it can rearrange any positions of cells without internal blocking as long as there is no timeslot conflict. In some cases, internal blocking may occur in the OTSI even though there is no timeslot conflict; then the OTSI is said to be blocking. The implementation complexity of the nonblocking OTSI is very high; thus, in practice, the blocking OTSI is a more attractive solution for performing timeslot interchange.

With reference to Figure 15.44, a feedback-buffered optical-packet switch based on the AWG device has been proposed by Chia et al. [38]. In this switch architecture, the switching plane is combined with N OTSIs, which are placed on the output side and are able to feed the delayed packets back to the input side of the switch. In the center, an AWG switch fabric is employed. Each inlet of the AWG switch fabric is associated with a tunable wavelength converter (TWC). The TWCs are needed because AWG devices switch optical signals according to their wavelengths. Those packets that lost contention are assigned delay

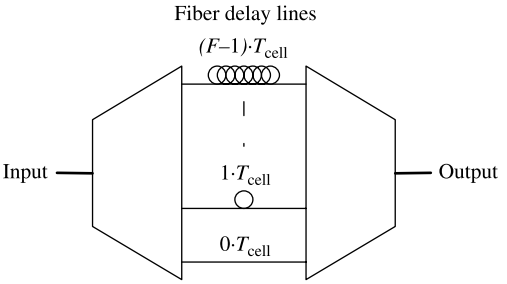


Figure 15.42 Nonblocking OTSI.

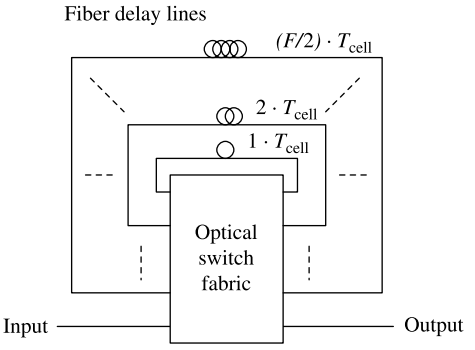
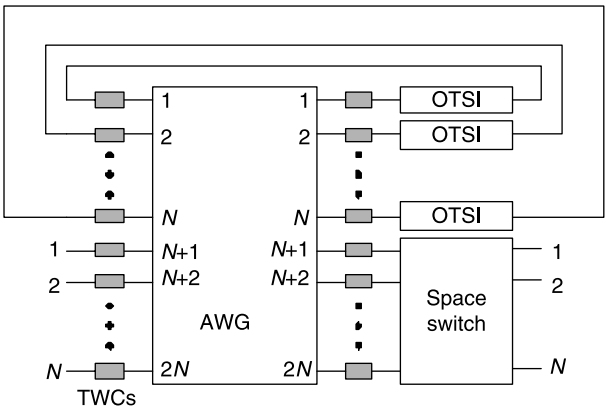


Figure 15.43 Blocking OTSI.



TWC : Tunable wavelength converter

OTSI: Optical timeslot interchanger

Figure 15.44 Optical buffered AWG packet switch.

values and switched to the proper OTSIs for buffering. Such a buffered switch is able to provide a low loss rate and low average delay. A major problem, however, is that TWCs are expensive. This problem can be resolved if other switching technologies, such as the micro-electro-mechanical system (MEMS) and SOA, are employed. Scheduling algorithms that can efficiently assign delay routes for optical packets by using blocking OTSIs have not yet received enough attention [38].

Time sliced optical burst switching (TSOBS) [37] is a variant of optical burst switching (OBS), in which burst contention is resolved in the time domain rather than in the wavelength domain, thus eliminating the necessity for wavelength conversion that occurs in the traditional OBS schemes [39, 40]. Ramamirtham and Turner [37] have also proposed an efficient scheduling algorithm for the per-input-OTSI optical switch. The architecture of the per-input-OTSI optical switch is given in Figure 15.45 where the OTSIs are the blocking ones as shown in Figure 15.43. In this algorithm, the existing schedule (switch configuration) is formulated as a directed graph, which gives all possible delay paths for data bursts. The assignment problem can thus be formulated as a searching problem in the directed graph. Nevertheless, there are two major problems with the per-input-OTSI switch. First, since an OTSI is employed and dedicated for each input port, the FDL requirement of the

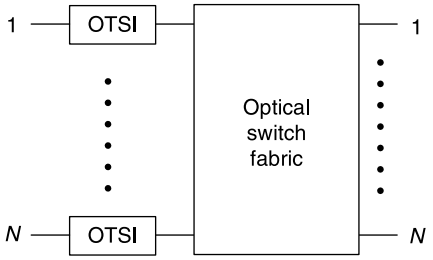


Figure 15.45 Per-input OTSI optical switch.

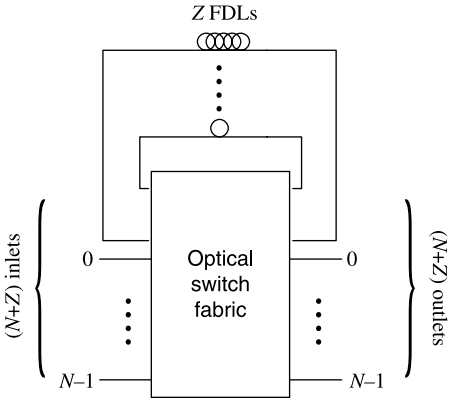


Figure 15.46 Single-stage shared-FDL optical switch.

entire switch is undesirably high. Second, as is the nature of an input-buffered switch, the switching schedule must resolve not only the output-port, but also the input-port contentions, thus limiting the overall performance of the switch.

In [41], Karol has proposed a single-stage shared-FDL switch for optical packet ATM switch. The structure of the single-stage shared-FDL switch is given in Figure 15.46. The switch contains a number of feedback FDLs that are shared among all input ports. Assume that there are Z feedback FDLs, N input ports, and N output ports. Each FDL can delay cells by a fixed number of timeslots and any two FDLs may have the same or different delay values. The outputs (inputs) of FDLs and the inputs (outputs) of the switch are collectively called the inlets (outlets) of the switch fabric, yielding $N + Z$ inlets and $N + Z$ outlets.

To resolve contention, Karol has also proposed a non-reservation scheduling algorithm for the single-stage shared-FDL switch in which he assumed specifically that the delay values of the Z FDLs are all different from 1 T_{cell} to Z T_{cell}. The algorithm is said to be non-reservation because there is no reservation (hence no departure time scheduling) for the cells that have lost in the contention and need to be buffered. That is, in each timeslot, cells can only be matched with the output ports for the current timeslot. For those buffered cells, there is no guarantee that they can obtain access to the desired output ports after coming out from the FDLs. Therefore, they may need to face another round of contention. Minimum reservation can be achieved by giving a higher priority to the cell that comes out from the longer FDL when resolving contention. However, since departure time is not scheduled in advance, the delay bound of Karol's algorithm can be very large and it may require a cell to be switched and recirculated many times. For example, the maximum number of recirculations required is ten in the simulation [41]. This is undesirable because the optical signal gets attenuated each time they are switched. Another issue of Karol's algorithm is that it is of high time complexity for scheduling cells, which is $O(Z^2)$, due to its sequential nature.

In the next section, we focus on the reservation scheduling algorithms in the single-stage shared-FDL switch. In contrast to the non-reservation scheduling algorithms, the reservation scheduling algorithms perform not only output port matching for the current timeslot, but also the FDL assignment for the entire journey of a delayed cell so that it can be scheduled to match with the desired output port in a future timeslot. The FDL assignment may involve one or more than one FDL circulation. If it is successful, the output port at the corresponding timeslot as well as the FDL path(s) along the journey are said to be

reserved for the cell. However, if a cell that needs to be delayed fails to be scheduled to a future timeslot for the desired output port owing to FDL and/or output-port conflicts, it is discarded without entering the switch so that it does not occupy any resources. To achieve low cell loss rate, two new algorithms for scheduling cells in the single-stage shared-FDL optical switch have been proposed by Lieu et al. [42]. They are: (i) the sequential FDL assignment (SEFA) algorithm, which searches FDL routes for cells on a cell-by-cell basis; and (ii) the multi-cell FDL assignment (MUFA) algorithm, which uses sequential search to find FDL routes for multiple cells simultaneously.

In addition to FDL and output port reservation, these scheduling algorithms also exhibit flexible features that allow switch designers to select the maximum delay value, say $F - 1$ timeslots, and the maximum number of FDL circulations, say K circulations, that can be imposed on a cell at the switch. This is very important from the traffic engineering point-of-view. Compared with Karol's FDL setting [41], these also allow two FDLs to have the same delay value, and assume that the delay values of FDLs are distributed among $2^0\,T_{\text{cell}}$, $2^1\,T_{\text{cell}}, \ldots, 2(f-1)\,T_{\text{cell}}$, where $f = \log_2 F$ for implementation. It is also worth noting that the single-stage shared-FDL switch and the proposed algorithms are also applicable for the circuit-based timeslot-wavelength division multiplexing (TWDM) networks [43–45] to perform timeslot interchange at a switching node so as to increase call admission rate.

15.5.2 Sequential FDL Assignment (SEFA) Algorithm

In SEFA [42], the FDL assignment is considered for a single cell at any given time. In practice, cells may arrive in the same timeslot. In that case, SEFA schedules these cells one after another. For forwarding cells, each shared-FDL switch maintains a configuration table. The configuration table is used to indicate the switching schedule of the switch, and it can be formulated into a slot transition diagram that includes all possible FDL routes for cells. The configuration table and the slot transition diagram are described as follows.

With reference to Figure 15.47, for $s \in$ outlets, $t \geq 0$, the entry of row s and column t in this table consists of two variables, $u(s, t)$ and $v(s, t)$, where $u(s, t) \in 0, 1$, and $v(s, t) \in$ inlets. Variable $u(s, t)$ is called the availability bit of outlet s and it is a Boolean variable that indicates whether outlet s is available in timeslot t. That is, if $u(s, t) = 1$, then outlet s is idle in timeslot t; if $u(s, t) = 0$, then outlet s is connected to inlet $v(s, t)$ in timeslot t. For the example given in Figure 15.47, $u(\text{output } p, t) = 0$ and $v(\text{output } p, t) = \text{input } 7$, indicates that output p is busy in timeslot t because it is scheduled to be connected to input 7. Note that the depth, that is, the number of columns, of the configuration table is F. The configuration table can be logically represented by a slot transition diagram, G, as shown in Figure 15.48.

In G, timeslot t (i.e., column t in the switch configuration table) is represented by a node with label $T(t)$. If FDLa is available in timeslot t (i.e., $u(\text{FDL}a, t) = 1$), it is represented by an arc from $T(t)$ to $T(t + Da)$, where Da is the delay value of FDL a. With such a representation, an available FDL route from timeslot t to timeslot τ is denoted by a path from $T(t)$ to $T(\tau)$ in G, where $\tau > t$. For example, path $T(t) \rightarrow \text{FDL}a \rightarrow T(t + 2) \rightarrow \text{FDL}c \rightarrow T(t + 6)$ in Figure 15.48 represents an available FDL route that can route a cell from timeslot t to timeslot $t + 6$. Note that since different FDLs can have the same delay values, G is a directed multigraph as shown in Figure 15.48.

For each node in G to match the desired output port with the cell request, node $T(t)$ also keeps the availability status of all outputs in timeslot t, that is, $u(\text{output } p, t)$ for all p, $0 \leq p \leq (N - 1)$. Therefore, finding a valid FDL route starting from timeslot t to a timeslot

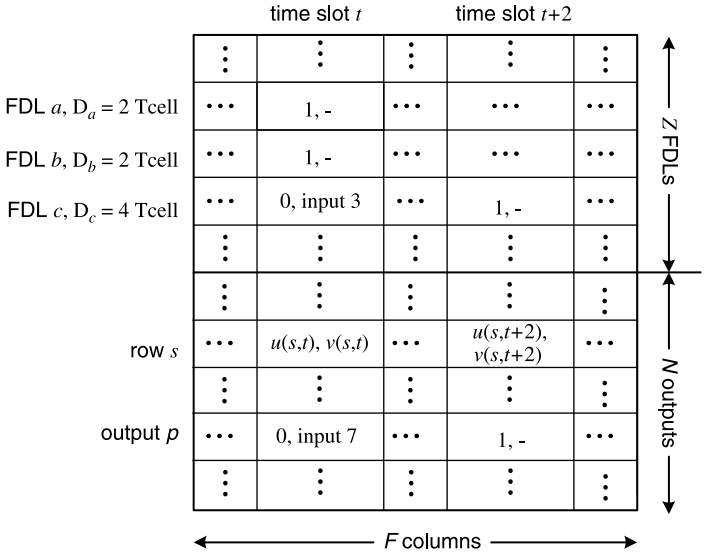


Figure 15.47 Switch configuration table of optical cross connect (OCX).

in which output p is available, is equivalent to finding a path in G from $T(t)$ to any $T(\tau)$ in which $u(\text{output } p, \tau) = 1$. Note that if $u(\text{output } p, t) = 1$ in the beginning, the cell can be routed to output p immediately in timeslot t without passing through any FDL.

An example is given below. Consider a shared-FDL switch with two inputs, two outputs and two FDLs as shown in Figure 15.49. Suppose that some routes have been previously scheduled and the scheduled switch configurations for the next four time slots are given in Figure 15.50. These scheduled configurations are stored in a configuration table as shown in Figure 15.51, and this table can be further represented by a slot transition diagram as shown in Figure 15.52.

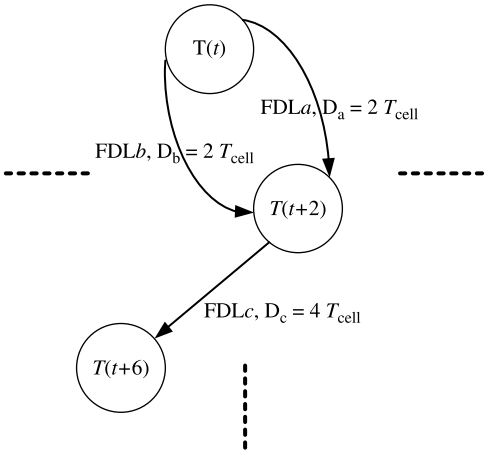


Figure 15.48 Slot transition diagram of OCX.

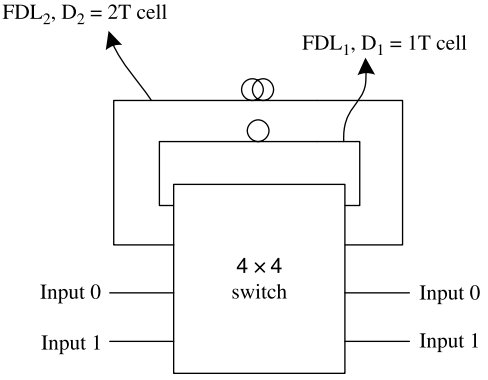


Figure 15.49 2×2 switch module, 2 FDLs.

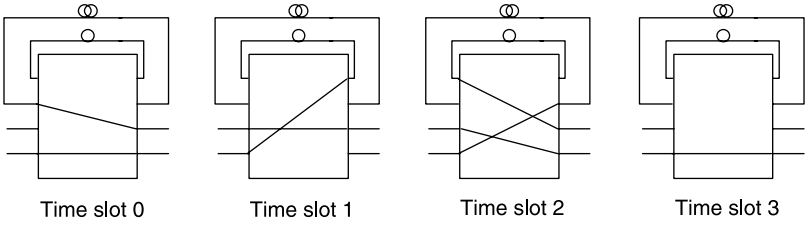


Figure 15.50 Scheduled connections in the switch module ($F = 4$).

With the above existing configuration, suppose that a new cell arrives from input 0 in timeslot 0, requesting to be connected to output 0. Since output 0 is not available in timeslot 0 (i.e., $u(\text{output } 0, 0) = 0$), we have to find an FDL assignment to route the new connection to any timeslot, say τ, in which $u(\text{output } 0, \tau) = 1$. With the slot transition diagram given in Figure 15.52, we can modify any search-based algorithms, such as the breadth-first search, for this objective. For the above example, two nonblocking FDL routes, $T(0) \rightarrow \text{FDL1} \rightarrow T(1) \rightarrow \text{FDL2} \rightarrow T(3)$ and $T(0) \rightarrow \text{FDL2} \rightarrow T(2) \rightarrow \text{FDL1} \rightarrow T(3)$ can be found, and both route the cell to $T(3)$ – the first timeslot in which output 0 is available.

Each time when a cell is passing through an FDL, regardless of the delay value of that FDL, it is said to be taking a delay operation (circulation). For example, both FDL routes given in the previous example impose two delay operations on the cell. Note that, a delay

	time slot 0	time slot 1	time slot 2	time slot 3	
FDL_1, D_1 = 1 Tcell	1, –	0, input 1	1, –	1, –	-----
FDL_2, D_2 = 2 Tcell	1, –	1, –	0, input 1	1, –	-----
output 0	0, FDL_2	0, input 0	0, FDL_1	1, –	-----
output 1	0, input 1	1, –	0, input 0	0, input 1	-----

Figure 15.51 Configuration table of the example.

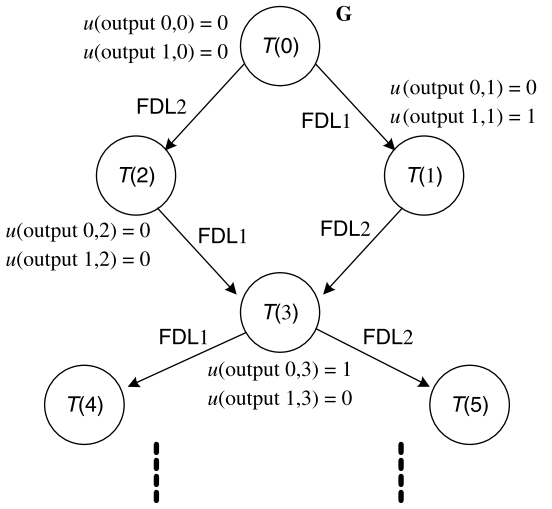


Figure 15.52 Slot transition diagram of the example.

operation is always followed by a switching operation. In practice, there may be a need to reduce the number of delay operations when scheduling FDL routes because optical signals are attenuated each time they are switched. Therefore, when multiple choices exist in which the desired output port is available, the selection criteria would be as follows:

1. Select the FDL route that involves the minimum number of delay operations.
2. If there are multiple valid FDL routes that involve the same minimum number of delay operations, we select the one with the smallest delay.
3. If there are multiple FDL routes involved in the same minimum number of delay operations and the same minimum delay value, one of them can be selected at random.

Theoretically, the size of the slot transition diagram can be infinitely large. However, one may limit the maximum number of delay operations (i.e., K operations) and/or the maximum cell delay (i.e., $F - 1$ cell times) in SEFA for practical implementation. Under these constraints, if neither a direct connection nor a FDL route can be assigned to a cell for the desired output port, the cell is discarded immediately without entering the switch.

SEFA Performance and Complexity. In the following, we provide the simulation results of SEFA for a 32×32 switch with 32 shared FDLs. Consider that the delay values of the 32 FDLs are distributed as evenly as possible between $1, 2, \ldots, 2^l, \ldots, F/2$ cell times, starting from 1. For instance, if $F = 128$, there are 5, 5, 5, 5, 4, 4, and 4 FDLs with delay values 1, 2, 4, 8, 16, 32, and 64 cell times, respectively. Compared with Karol's FDL length selection in [41], the reasons why an exponentially increasing length rather than linearly increasing length is chosen are as follows. (1) For those cells that need a large delay value for the desired output ports, we can delay them with only a few delay operations. (2) With fewer choices of FDL length, the size of slot transition diagram is smaller; hence the complexity of the algorithm can be reduced. The Bernoulli arrival process is assumed. Furthermore, when there is a cell arriving at an input port, it is equally likely to be destined for any one

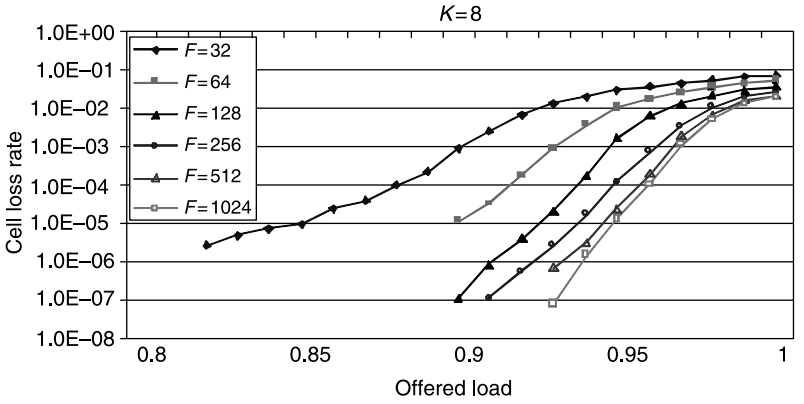


Figure 15.53 Cell loss rate versus offered load (SEFA).

of the output ports. If multiple cells arrive at the switch at different input ports in the same timeslot, they are scheduled one after another sequentially.

Figure 15.53 shows the cell loss rate and the average cell delay, respectively, of cells with respect to the offered load under different F values in SEFA, assuming $K = \infty$. From Figure 15.53, the larger the value of F, the smaller the cell loss rate. The reason is twofold. (1) The larger F provides more alternative delay values for cells to search for timeslots in which the desired output ports are available. (2) An FDL with delay value D can "buffer" up to D cells; thus, according to the above setting of FDL delay values, the larger F implies a larger buffer size for buffering cells. It is worth noting that when $F = 128$, the cell loss rate is $\sim 10^{-7}$ at a load of 0.9.

However, a large F is not always preferable. With reference to Figure 15.54, the average cell delay increases as F increases. The reason is that, in SEFA we always select the FDL route that involves the minimum number of delay operations when multiple choices are available. In the worst case, the FDL route with one delay operation having a delay value of $F/2$ will be selected rather than the FDL route with two delay operations but having a total delay value of only two. Thus it causes a larger cell delay. One possible way of resolving

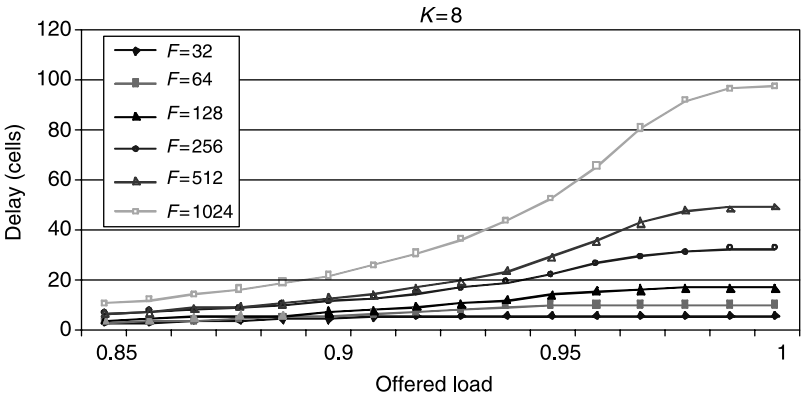


Figure 15.54 Average cell delay versus offered load (SEFA).

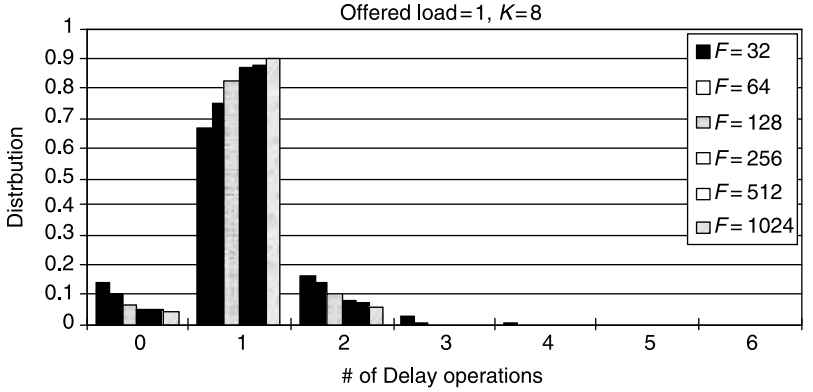


Figure 15.55 Distribution of the number of delay operations.

this problem is to search for the FDL routes in ascending order according to the total delay. However, this may result in more delay operations.

We also obtained the distribution of the number of delay operations under different F values in SEFA, assuming $K = \infty$ and offered load $= 1$. Figure 15.55 shows that most cells passing through FDLs require less than four FDL delay operations. The reason is that, the more delay operations a FDL route involves, the more chance there is that it incurs FDL contention. Thus it can be assigned to a cell successfully with a lower probability. It is worth noting that, when $F \geq 128$, no cell is assigned a FDL route of four or more delay operations.

From the above observation, it may be conjectured that SEFA performs equally well when $K = 2$ and $K = \infty$, assuming $F \geq 128$. To verify this argument, Figure 15.56 shows the simulation result of cell loss rate with respect to the offered load under different settings of K, where F is set to 128. Basically, when $K = 2$, $K = 3$, and $K = \infty$, respectively, the curves coincide with each other with slight differences, thus it verifies the above argument.

To find the time complexity of SEFA, we assume that the time needed for the SEFA scheduler to access a node in graph G is C_{se}. For instance, $C_{se} = 5$ ns if the clock rate of

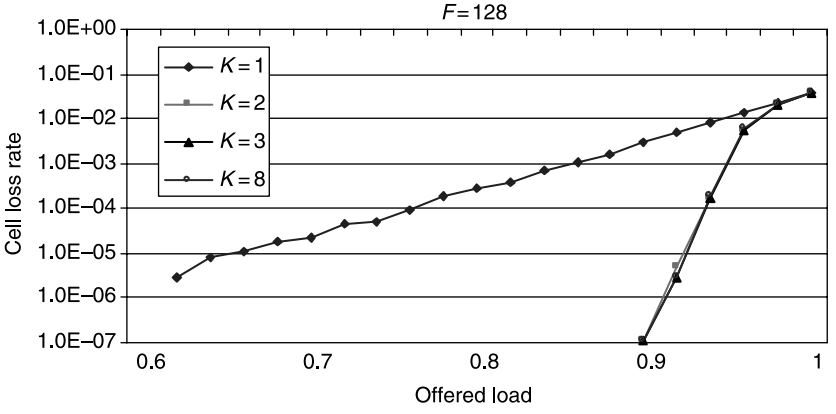


Figure 15.56 Cell loss rate under different settings of K.

the scheduler is 200 MHz. The scheduler in the worst case needs to search over all nodes in G in order to find out a FDL assignment for a cell, so the number of nodes in G is another factor that contributes to the time complexity. Let Q be the number of nodes in G. There is no close-form solution to the value of Q in terms of F and K, but it can be observed that Q grows exponentially with K. Fortunately, if we set $K = 2$ and $F = 128$, the performance is satisfactory enough as shown in Figures 15.53 and 15.54, (yielding $Q = 36$ only). Finally, since at most N cells can arrive at the switch in a particular timeslot, the time complexity of SEFA is $N \times Q \times C_{se}$. For instance, if $N = 32$, $Q = 36$, and $C_{se} = 5$ ns, the time complexity of SEFA is $32 \times 29 \times 5\,\text{ns} = 4.64\,\mu\text{s}$.

15.5.3 Multi-Cell FDL Assignment (MUFA) Algorithm

In MUFA [46], we consider the FDL assignment for multiple cells simultaneously. To guarantee that the FDL routes with fewer delay operations are searched and assigned for cells earlier, the slot transition diagram is modified as follows.

With reference to Figure 15.57, in the modified slot-transition diagram, the node that represents the current timeslot is called node $T_0(0)$, where the subscript denotes the level of the node. $T_0(0)$ is the only level-0 node. Consider $1 \le k \le K$, where K is the maximum number of delay operations for a cell. A node $T_k(t)$ is said to be a level-k node, if any cell arriving at the switch at $T_0(0)$ can take k delay operations to reach the node, where t is the total delay value of the FDLs traversed. For example, $T_2(3)$ is a level-2 node because a cell can traverse two FDLs, with delay values 1 and 2 (total delay = 3), respectively, from $T_0(0)$ to $T_2(3)$. Note that for a delay value, there can be different nodes at different levels, such as $T_1(2)$ and $T_2(2)$ in Figure 15.57.

The parent of a level-k node is a level-$(k-1)$ node. A node can have multiple parents and/or multiple children. The link from a parent to a child is unique, and it indicates that there are FDLs that can delay cells from the parent to the child (delay value is 2^l, where $0 \le l \le 6$ in this example), regardless of whether these FDLs are available in the parent node. However, the availability status of all output ports and FDLs at timeslot t can be tracked with the availability bits that are kept locally at $T_k(t)$.

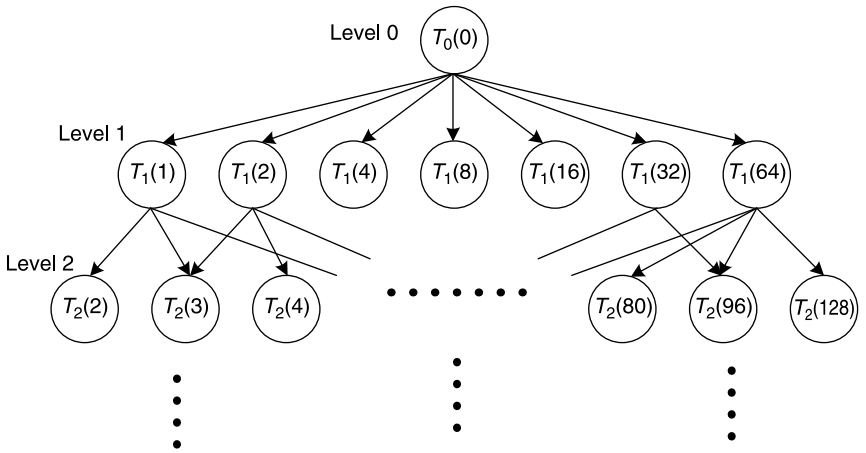


Figure 15.57 Modified slot transition diagram.

Let $OPA_j(t)$ denote the availability bit of output port j and $FA_q(t)$ denote the availability bit of FDL q in timeslot t, where $0 \leq j \leq N-1$ and $0 \leq q \leq Z-1$. That is, if $OPA_j(t) = 1$ ($FA_q(t) = 1$), then output port $j(FDL_q)$ is available in timeslot t; if $OPA_j(t) = 0$ ($FA_q(t) = 0$), then output port $j(FDL_q)$ is busy in timeslot t.

In any timeslot, the shared-FDL switch may receive up to N requests. Upon receiving these requests, the switch controller activates the MUFA algorithm, which consists of $K+1$ levels of assignment from level 0 to level K. The level 0 assignment has only one step, in which the algorithm attempts to assign direct connections for cells. The requests that have not been assigned connections (due to contention) are called the unfulfilled requests. Unfulfilled requests can be granted via FDL routes. This is done in the higher levels of assignment.

For $1 \leq k \leq K$, in the level-k assignment, the MUFA algorithm tries to assign level-k routes (routes from $T_0(0)$ to level-k nodes) for the unfulfilled requests. With reference to Figure 15.58, the grant decisions of level-k nodes for the unfulfilled requests are made by their parent node (which is a level-$(k-1)$ node). Moreover, at any time, only one parent node performs the granting process. For example, during the level-1 granting process, node $T_0(0)$ is the parent of all level-1 nodes and it makes grant decisions for its children; during the level-2 granting process, each level-1 node acts as a parent node, one after another, from $T_1(1)$ to $T_1(64)$ to make grant decisions for their children (level-2 nodes), and so on. The granting process of a parent node is described as follows.

With reference to Figure 15.59, to make a granting decision, the parent node needs to collect three sets of data in addition to the FDL availability status of itself:

1. Output port availability status from all its children.
2. Available FDL routes from $T_0(0)$ (such information is kept at the ancestor nodes, updated in the end of each iteration, and is passed to the parent node in question when necessary).
3. Unfulfilled requests from the last processing node.

During the granting process, the parent node matches the unfulfilled requests with the output-port availabilities (OPAs) of its children in accordance with the number of available FDL routes from $T_0(0)$ to the child node. If multiple choices are available for an output port, the parent chooses the child node with the smallest delays for the corresponding requests. With reference to Figure 15.60, after the granting process, the parent node sends the updated

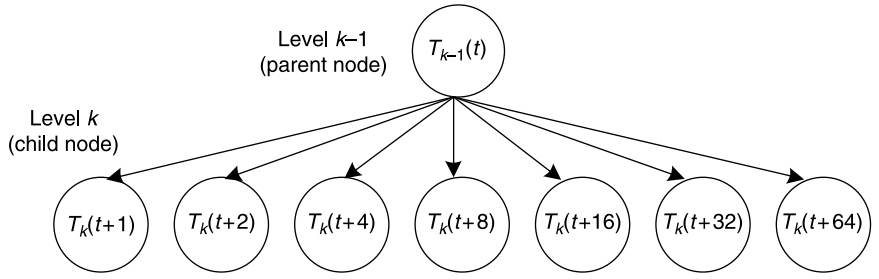


Figure 15.58 Parent node makes granting decisions for the child nodes.

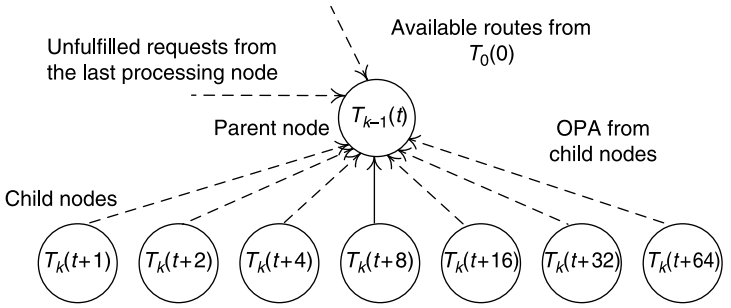


Figure 15.59 Before granting process at a parent node.

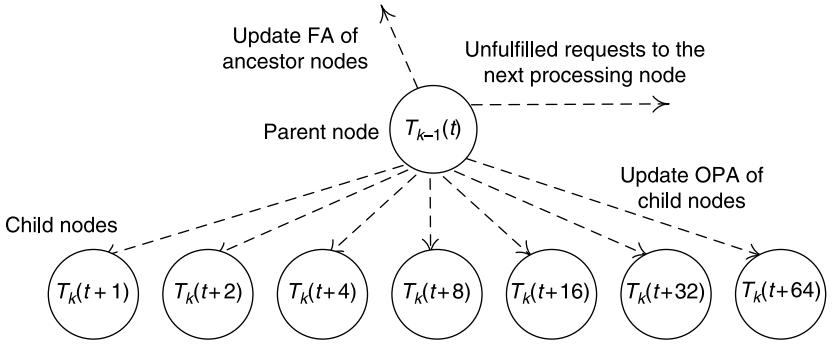


Figure 15.60 After granting process at a parent node.

OPAs back to the child nodes, updated FDL availabilities (FAs) back to the ancestors, and the unfulfilled requests to the next processing nodes. After level-k granting is done (i.e., all level-$(k-1)$ nodes have acted as parent nodes), the algorithm proceeds to level-$(k+1)$ until $k = K$.

An example of MUFA is given below. Consider a shared-FDL switch with four inputs, four outputs and four FDLs that have delay values of 1, 1, 2, 2, respectively, as shown in Figure 15.61. Suppose that at timeslot 0, four packets arrive and they are all destined for output 3, where the packet from input i to output j is denoted by (i, j). A modified slot-transition diagram can be constructed as shown in Figure 15.62. Note that this diagram is independent of the packet arrival.

To schedule the above packets, in level-0 assignment, $T_0(0)$ can grant one of the four packets (say packet $(0, 3)$) for output 3 at timeslot 0. This process also turns $OPA_3(0)$ from 1 to 0 in such a way that no other packets can be matched with output 3 at timeslot 0. For the other three packets, they need to be buffered in the FDLs.

With reference to Figure 15.63, in level-1 assignment, $T_0(0)$ acts as the parent of, and makes granting decisions for, $T_1(1)$ and $T_1(2)$ simultaneously. Since all FDLs are available at the time (i.e., $FA_q(0) = 1$ for all FDL_q, $1 \le q \le 4$), two of the packets [say $(1, 3)$ and $(2, 3)$] can be assigned the FDLs with delay values 1 and 2 (say FDL1 and FDL3), for output 3 in timeslots 1 and 2, respectively. This process also turns $FA_1(0)$, $FA_3(0)$, $OPA_3(1)$, and $OPA_3(2)$ from 1 to 0.

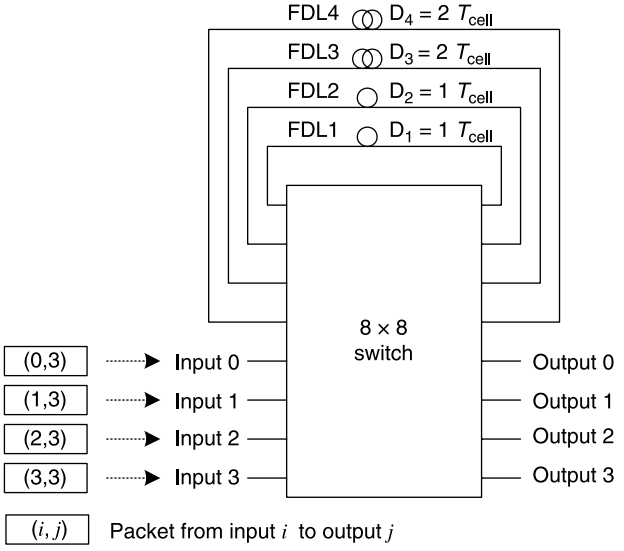


Figure 15.61 4 × 4 switch module, 4 FDLs.

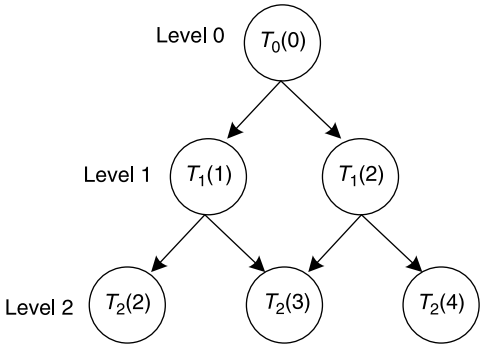


Figure 15.62 Modified slot-transition diagram for a 4 × 4 switch.

The remaining unfulfilled packet, (3, 3), can be granted in level-2 assignment, in which $T_1(1)$ first acts as the parent of $T_2(2)$ and $T_2(3)$, and then $T_1(2)$ acts as the parent of $T_2(3)$ and $T_2(4)$. With reference to Figure 15.64, consider $T_1(1)$ acting as the parent of $T_2(2)$ and $T_2(3)$. From the top of $T_1(1)$, since there remains only one FDL (i.e., FDL2, where $FA_2(0) = 1$) that can shift a packet from $T_0(0)$ to $T_1(1)$, $T_1(1)$ can grant at most one packet for either $T_2(2)$ or $T_2(3)$. To the children of $T_1(1)$, output 3 at timeslot 2 has been assigned to another packet (i.e., $OPA_3(2) = 0$). Therefore, $T_1(1)$ can only grant packet (3, 3) for output 3 at $T_2(3)$, and this consumes a FDL with delay value 2 (say FDL3) at timeslot 1. This process also turns $FA_2(0)$, $FA_3(1)$, and $OPA_3(3)$ from 1 to 0. The entire scheduling result of MUFA is given in Figure 15.65.

MUFA Performance and Complexity. There are two possible phenomena that make MUFA and SEFA perform differently. (1) In MUFA, for a particular output port, it is

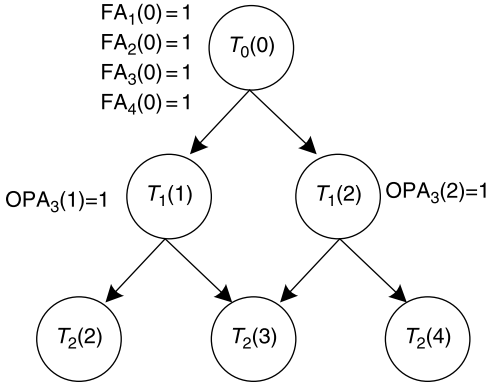


Figure 15.63 $T_0(0)$ acts as the parent of $T_1(1)$ and $T_1(2)$.

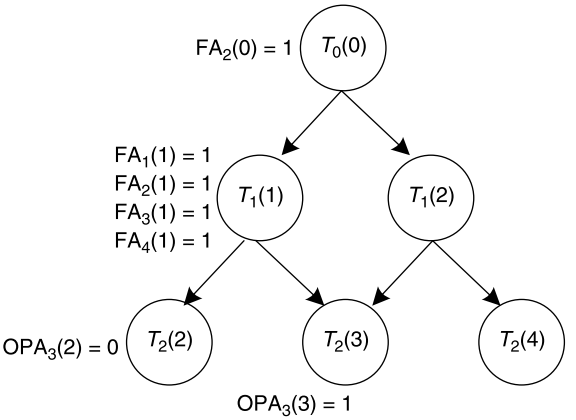


Figure 15.64 $T_1(1)$ acts as the parent of $T_2(2)$ and $T_2(3)$.

guaranteed that the FDL routes with the fewer delay operations are assigned earlier. However, considering two FDL routes with the same number of delay operations, it is possible that the route with the larger delay is selected rather than the route with the smaller delay. This occurs when the former's parent node has a smaller index than that of the latter's parent node. For instance, with reference to Figure 15.57, the delay of the FDL route of $T_0(0) \rightarrow T_1(32) \rightarrow T_2(96)$ is larger than that of the FDL route of $T_0(0) \rightarrow T_1(64) \rightarrow T_2(80)$. Such a phenomenon does not occur in SEFA. (2) In SEFA, since cells that could be destined for different outputs are scheduled sequentially, it is possible that FDL routes with more delay operations are assigned to cells in the early time in such a way that they occupy the FDL resources and prevent the subsequent cells from finding FDL routes with fewer delay operations. In this case, FDL resources are less efficiently used in SEFA than MUFA. From Figures 15.66 and 15.67, phenomenon (1) makes SEFA perform better at a load < 0.95; phenomenon (2) makes MUFA perform better at a load > 0.95. Overall, the difference of performance between SEFA and MUFA is not significant.

To find the time complexity of MUFA, with reference to Figures 15.59 and 15.60, suppose that the time needed for a parent node to collect the necessary information is C_{cl}; the time

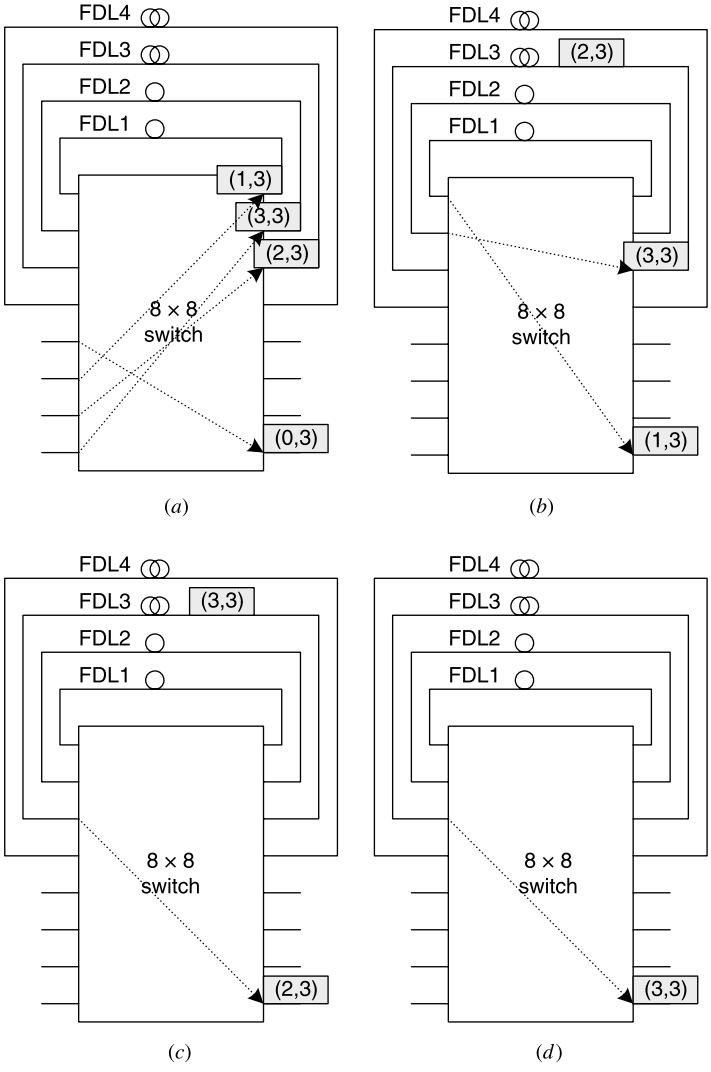


Figure 15.65 Scheduling result of MUFA. (*a*) Routing in Timeslot 0; (*b*) Routing in Timeslot 1; (*c*) Routing in Timeslot 2; (*d*) Routing in Timeslot 3.

needed for the parent node to grant unfulfilled requests is C_{gr}, where C_{gr} includes a step of parallel AND operations (to match the unfulfilled requests with the available output ports at each child node) and $\log_2 F$ sequential steps of bit comparison (to grant the matches for each child node); the time needed for the parent node to pass the necessary information to the corresponding nodes is C_{ps}. Moreover, let P be the number of nodes that act as parent nodes during the MUFA process. The time complexity of MUFA is $P \times (C_{cl} + C_{gr} + C_{ps})$. P is a function of K and F. For example, when $K = 2$ and $F = 128$, P is equal to $1 + 7 = 8$. Let us assume $C_{cl} = C_{ps} = 5\,\text{ns}$, then $C_{gr} = (1 + \log_2 F) \times 5\,\text{ns} = 40\,\text{ns}$, and the time complexity is $8 \times 50\,\text{ns} = 400\,\text{ns}$.

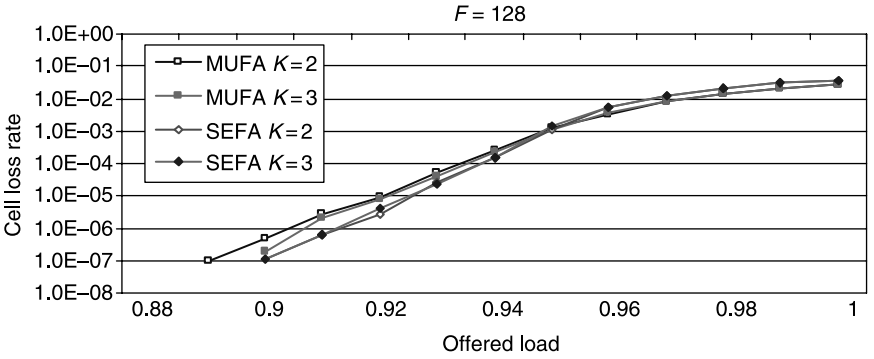


Figure 15.66 Cell loss rate versus offered load (MUFA).

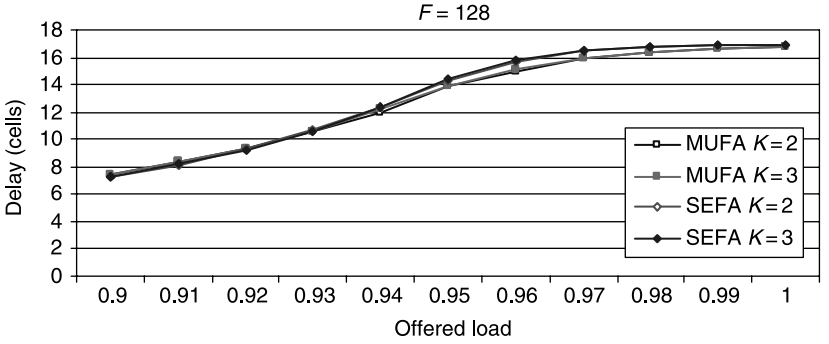


Figure 15.67 Average cell delay versus offered load (MUFA).

15.6 ALL OPTICAL PACKET SWITCH WITH SHARED FIBER DELAY LINES – THREE-STAGE CASE

The scalability of the single-stage shared-FDL switch is greatly limited by the number of required cross points, which is $(N + Z)^2$. To further enhance the scalability of the optical-buffered switches, it is common to consider the multi-stage modular switch architecture due to its high scalability and low complexity nature. Among all multi-stage modular switch architectures, the Clos-network is the most practical and frequently used scheme and gives a balance of switch performance and hardware complexity.

A three-stage optical Clos-network switch (OCNS) contains $KN \times M$ input modules (IMs), $KM \times N$ output modules (OMs) and $MK \times K$ center modules (CMs). Note that the size of the switch is $NK \times NK$. To buffer cells when contention occurs, FDLs can be placed at IMs, CMs, and/or OMs. However, different FDL placements can result in different scheduling complexity and performance. If the FDLs are only placed at OMs, cells are forced to be routed to the last stage as soon as they arrive at the switch. Since each OM can only accept up to M cells at any given timeslot, excess cells will be discarded, which results in a high cell-loss rate. If the FDLs are only placed at CMs, global availability information is needed when scheduling a batch of incoming cells and thus this placement discourages distributed FDL scheduling schemes. As a result, the focus is on the OCNS in

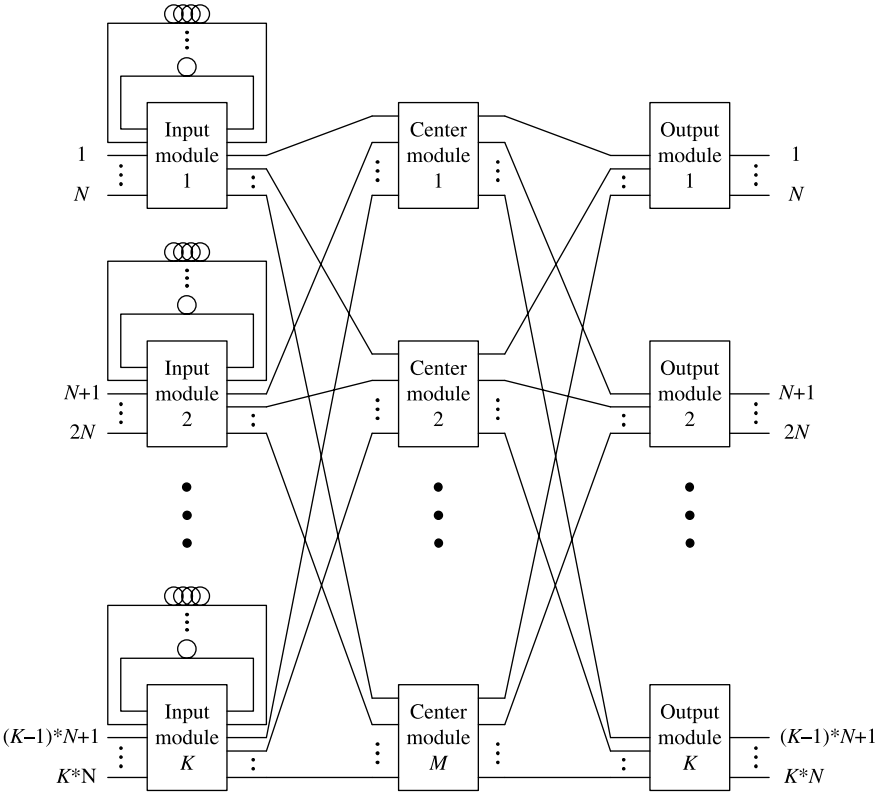


Figure 15.68 Three-stage shared-FDL-IM optical Clos-network switch.

which FDLs are only placed at IMs, as shown in Figure 15.68. We call this switch structure the three-stage shared-FDL-IM OCNS (SFI-OCNS). In the SFI-OCNS, cells can be delayed only at the first stage, while the second and third stages are used only for routing purposes; hence cells can be scheduled in a distributed manner. To give a fair comparison, we have studied the performance of different FDL placements. Our simulation results confirm that the SFI-OCNS has the lowest cell loss rate.

In addition to the departure schedule and FDL assignment, there is another issue in the three-stage SFI-OCNS: the central-route assignment. It is well-known that the number of CMs (i.e., M) in a three-stage Clos-network switch determines the non-blocking characteristic of the switch. If $M \geq 2N - 1$ [47], the switch is said to be strictly non-blocking because central routes can be arbitrarily assigned for the existing connections, yet none of the future connections will be blocked. However, given an M smaller than $2N - 1$, the central routes must be assigned carefully; otherwise rearrangement may be necessary or internal blocking may occur. There are two kinds of central-route assignment algorithms for Clos-network switches: the optimized and the heuristic. Although the optimized algorithms can always find the optimal solution to the central-route assignment problem, they have a high time complexity. Therefore, in practice, heuristic algorithms are preferable for scalability at a cost of slight performance degradation.

It is challenging to devise efficient scheduling algorithms to assign departure times, FDL routes, and central routes for cells in the three-stage SFI-OCNS. SEFA and MUFA

scheduling schemes in Sections 15.5.2 and 15.5.3, respectively, are extended for the SFI-OCNS. MUFAC is a practical algorithm to perform cell scheduling for the SFI-OCNSs due to its graceful scalability and distributed nature.

15.6.1 Sequential FDL Assignment for Three-Stage OCNS (SEFAC)

With reference to Figure 15.68, since each input module is a single-stage shared-FDL switch, it maintains its own slot transition diagram. In addition, the whole system has a bigger configuration table that keeps track of the availability of all outputs in each timeslot. Therefore, the output port and center-route availabilities are accessible by all input modules to perform scheduling algorithm. The FDL assignment and cell departure schedule are described in Section 15.6.2. Each input port takes turn to search for the earliest timeslot that satisfies the following three conditions: (i) that the destined output port is available in the timeslot, (ii) that there exists an FDL route on the corresponding input module that can move the cell from the current timeslot to the earliest timeslot, (iii) that a route between the IM and the destined OM is available at that timeslot. When all three conditions are met, the input port assigns the FDL routes, departure time, and randomly selects the center-route among all available center-routes. This searching process is performed one input port after another. In order to achieve fairness among all input ports, a round-robin mechanism can be included in the SEFAC algorithm in such a way that the priority of searching is rotated among all input modules.

15.6.2 Multi-Cell FDL Assignment for Three-Stage OCNS (MUFAC)

The three main tasks in the MUFAC algorithm are to:

1. Assign FDL routes in IMs.
2. Schedule cell departure times in correspondence with the output port availability.
3. Assign central-routes between IMs and OMs for multiple cells simultaneously.

In order to accomplish all three tasks, the original single-stage MUFA algorithm is enhanced and further incorporated with Karol's matching algorithm (Section 12.5).

Karol's Matching Algorithm. An example of the matching sequence in Karol's matching algorithm in the OCNS is given as follows. Let us consider a 9×9 three-stage SFI-OCNS, which has three IMs, three CMs, and three OMs. Therefore, it takes three mini-slots for all the IMs to finish performing Karol's matching with all the OMs. Figure 15.69 illustrates such a matching sequence.

To find an available center-route in Karol's algorithm is quiet simple. Each IM–OM pair can be connected via M CMs; a vector can be used for each input and output module to record the availability of the central modules. With reference to Figure 12.10, the A_i vector records the available route from IM_i to all CMs. Similarly, the B_j vector records the available route between all CMs and OM_j. Each element in those vectors corresponds to each CM; a "0" means available and "1" represents unavailable. For those pairs of modules that have a cell to dispatch between them, the two vectors will be compared to locate an available central module if any.

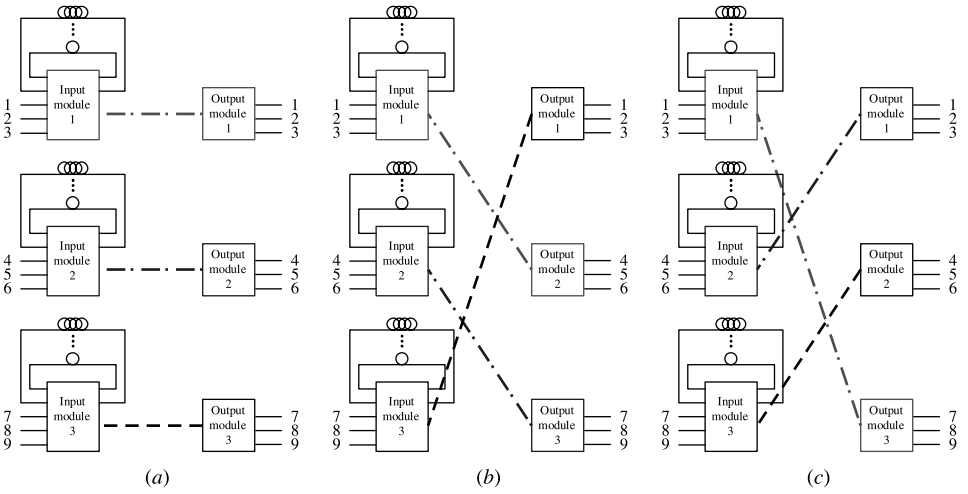


Figure 15.69 Karol's matching algorithm in three-stage SFI-OCNS: (*a*) First Mini-slot; (*b*) Second Mini-slot; (*c*) Third Mini-slot.

MUFAC. In MUFAC, each IM maintains its own transition diagram, and each level-k node T_k(t) (as shown in Fig. 15.57) keeps the FDL availabilities of that IM for timeslot t. In addition, each OM keeps the corresponding output-port availabilities (OPAs), and each of the IMs and OMs keep the corresponding central-route availabilities (CRAs). With the transition diagram, nodes take turn to be the parent node from the level-0 node to each of the level-$(L-1)$ nodes. Each of these turns is called an iteration.

Based on Karol's matching algorithm, an iteration is further divided into K cycles. In each cycle, each IM is paired up with a particular OM, yielding K IM–OM pairs, and only the cell requests for these IM–OM pairs will be handled. This is done by means of four phases, namely request, grant, accept, and update.

In the request phase, each IM works independently from the others, in which the parent node sends the unfulfilled requests to its child nodes so that they can execute the grant phase independently. At the same time, each child node also collects the OPAs from the paired OM for the corresponding timeslot so that it can grant the unfulfilled requests with the available output ports in the grant phase. After granting the unfulfilled requests, the child nodes pass their grant decisions back to the parent node. At the same time, the parent node collects the CRAs from its home IM and the paired OM. In the accept phase, the parent node makes the accept decision based on the following four criteria: (1) unfulfilled input requests; (2) availability of FDL on that IM for the corresponding timeslots; (3) availability of center-routes from that IM to the paired OM in the corresponding timeslots; and (4) when multiple grants occur, the parent accepts the grant with the earliest departure time. After the parent node makes the accept decision, it passes the decision to its child nodes for updating. In the update phase, the parent node updates the CRAs and FDL availabilities, while the child nodes update OPAs on the paired OM. This completes a cycle of MUFAC. Note that all these phases can be executed in a distributed manner. After K cycles, a node is done with the role of the parent node, and the next node will take the role and run again the K cycles. This process continues until the last parent node [a level-$(L-1)$ node] is done with the iteration.

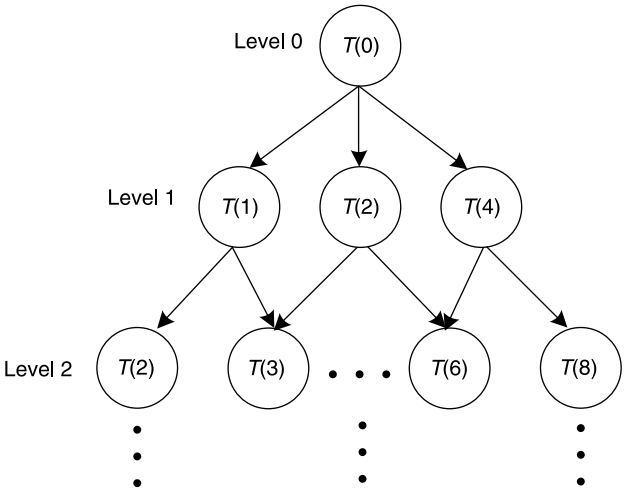


Figure 15.70 Slot transition diagram for a 9 × 9 OCNS.

A MUFAC example is given below. With reference to Figure 15.68, suppose that the switch just gets reset, so all output ports are available. We also assume the incoming cell requests from input 1 to input 9 are output ports 1, 4, 7, 1, 4, 7, 1, 4, 7, respectively. Each IM has a transition diagram as shown in Figure 15.70.

In the first iteration, MUFAC tries to assign direct connections for the cell requests. In this 9 × 9 three-stage OCNS, it requires three cycles to finish this task. Within each cycle, each IM consults with a different OM for current OPAs. In the first cycle, IM1 obtains OPAs and center-routes information from OM1. It finds out that output port 1 is available and assigns the direct connection. Similarly, IM2 and IM3 resolve output requests 4 and 7, respectively. In the second and third cycle, no more requests can be resolved because all desired output ports have been assigned in the first cycle. After three cycles, the assignment diagram for T0 nodes at each IM is shown in Figure 15.71.

In the second iteration, T0 becomes the parent of T1, T2, and T4. A four-step assignment process, namely, request, grant, accept, and update is performed three times in three cycles. In the first cycle, each IM does not have an unfulfilled request heading to the matched OM, so no assignment is made. In the second cycle, IM1 has a match with OM2 for output port 4 at T1; IM2 resolves output port 7 with OM3 at T1; and IM3 finds solution for output port 1 with OM1 at T1. In the third cycle, all remaining unfulfilled requests are resolved at T2. Figures 15.72 and 15.73 show the assignment diagram at T1 and T2 for each IM, respectively.

15.6.3 FDL Distribution in Three-Stage OCNS

The cell-loss performances of different FDL placement schemes for the three-stage OCNS are studied by using SEFAC in this subsection. We assume that the three-stage OCNS has 32 IMs, 32 OMs, 32 CMs, and each IM (OM) has 32 input ports (output ports). The overall switch size is 1024 × 1024. We consider five different cases of FDL placement scheme and compare their performances under uniform traffic. With reference to Figure 15.74, let Zin be the number of FDLs that are attached on each input module, and

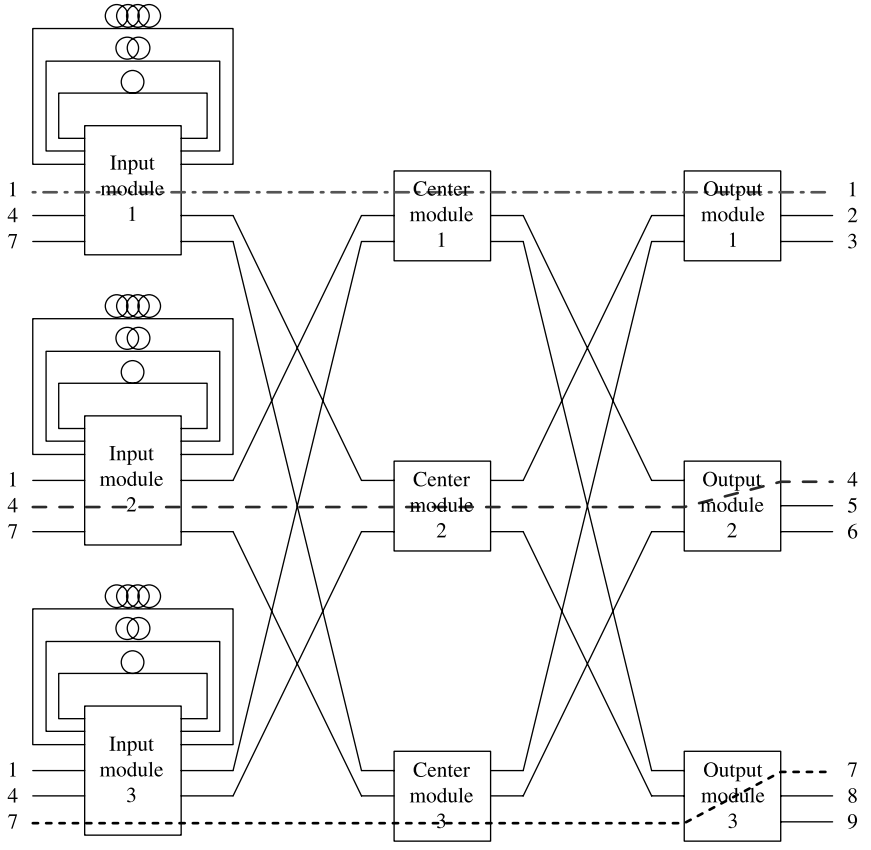


Figure 15.71 Assignment diagram for T0.

Zout be the number of FDLs that are attached on each output module. To give a fair comparison, we let Zin + Zout = 32. The cases studied are as follows: (*a*) Zin = 32, Zout = 0; (*b*) Zin = 24, Zout = 8; (*c*) Zin = 16, Zout = 16; (*d*) Zin = 8, Zout = 24; and (*e*) Zin = 0, Zout = 32.

As shown in Figure 15.75, placing all FDLs at the IMs achieves the best performance, while placing them all on the OMs results in the worst performance. To explain this, let us assume that there is no blocking in the middle stage and the entire switch is logically equivalent to a set of K independent concentrator–knockout switches [48], each having the structure as shown in Figure 15.76. Since each IM has no buffer, incoming cells in the IMs are forced to go through the center stage to the OMs immediately upon their arrival at the switch. In the worst case scenario, all $K \times N$ input ports could have cells destined for the same OM. However, at any given timeslot, only up to M cells can arrive at a given OM and the excess cells will be discarded by the CMs even before the cells reach the OM; this is the so-called knockout phenomenon. Therefore, the loss rate is the highest when all buffers are placed at the OMs. On the contrary, when FDLs are located at the IMs, cells can be buffered at the input stage and then directed to the corresponding OMs. Therefore, the performance is the best among all cases.

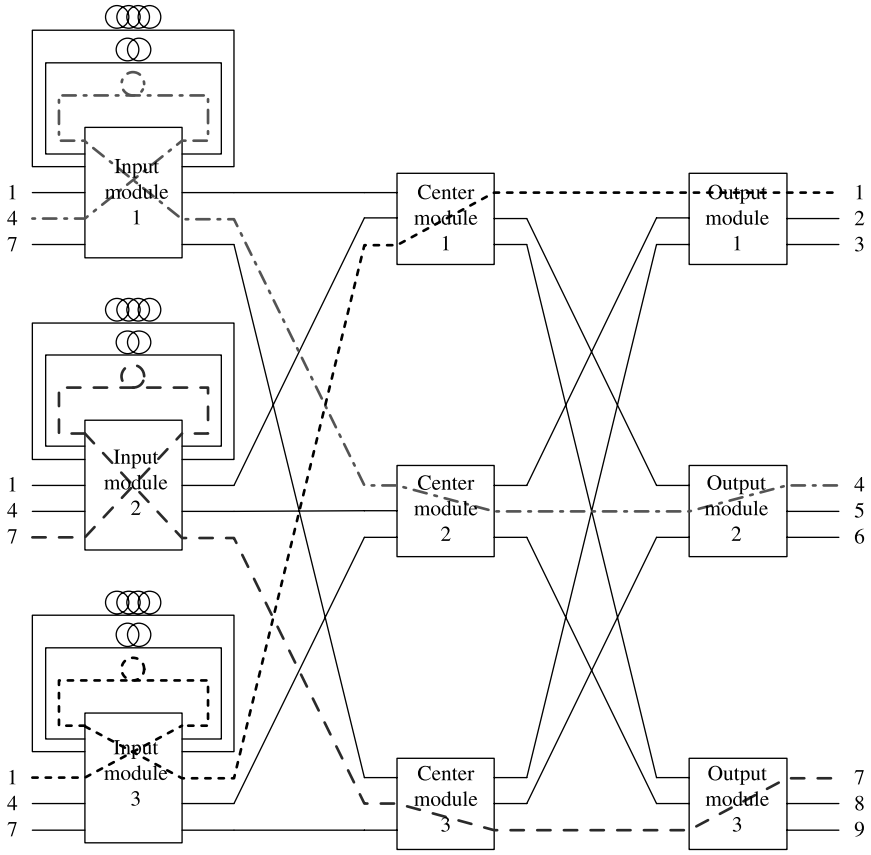


Figure 15.72 Assignment diagram for T1.

15.6.4 Performance Analysis of SEFAC and MUFAC

In our performance evaluation, we considered a 1024×1024 SFI-OCNS for both SEFAC and MUFAC. The SFI-OCNS consists of 32 IMs, 32 CMs, and 32 OMs, each module has 32 inputs and 32 outputs; we assume 32 FDLs are employed at each IM, and there are 5, 5, 5, 5, 4, 4, and 4 FDLs with delay values 1, 2, 4, 8, 16, 32, and 64 cell times, respectively. Furthermore, we limited the delay operation for each cell to 2 in both scheduling algorithms. In addition, we used a single stage 32×32 SEFA as a benchmark.

As shown in Figure 15.77, both three-stage SFI-OCNX FDL assignment algorithms can achieve 10^{-7} loss rate at 0.87 loads. There are two possible phenomena that make MUFAC and SEFAC perform differently. (1) In MUFAC, for a particular output port, we guarantee that the FDL routes with fewer delay operations are assigned earlier. However, considering two FDL routes with the same number of delay operations, it is possible that the route with the larger delay is selected rather than the route with the smaller delay. This occurs when the former's parent node has a smaller index than that of the latter's parent node. Such a phenomenon does not occur in SEFAC. (2) In SEFAC, since cells that could be destined for different outputs are scheduled sequentially, it is possible that FDL routes with the more delay operations are assigned to cells in the early time in such a way that

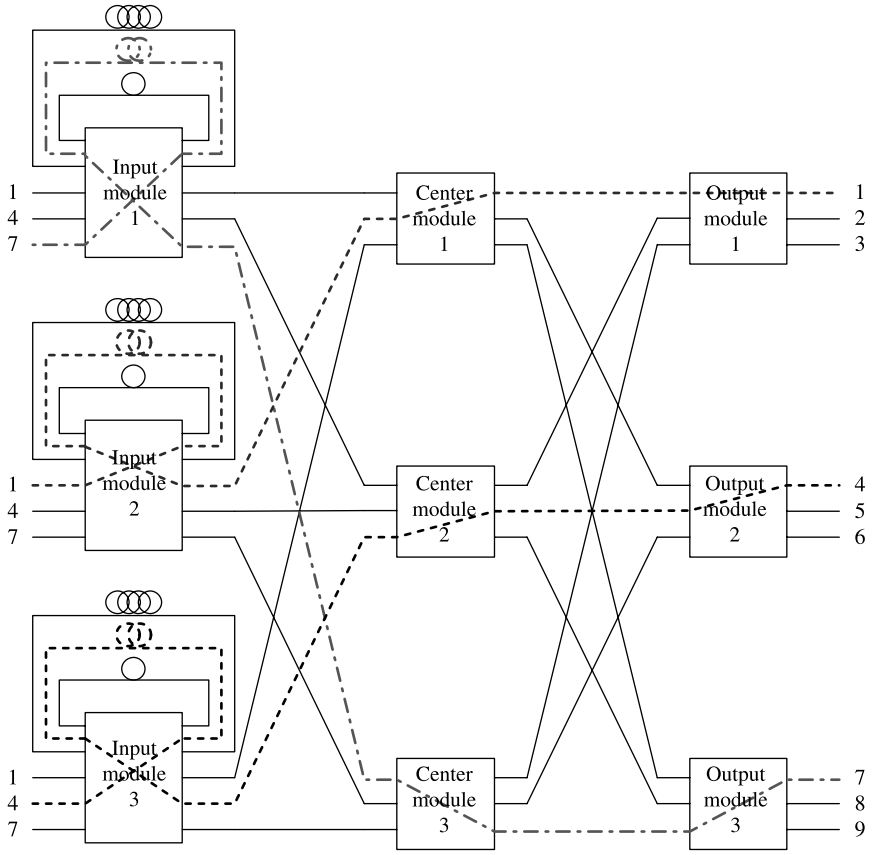


Figure 15.73 Assignment diagram for T2.

they occupy the FDL resources and prevent the subsequent cells from finding FDL routes with the fewer delay operations. In this case, FDL resources are less efficiently used in SEFAC than MUFAC. From Figure 15.77, phenomenon (1) makes SEFAC perform better at a load <0.94; phenomenon (2) makes MUFAC perform better at a load >0.94. Overall the performances of SEFAC and MUFAC for the SFI-OCNS are compatible.

Figure 15.78 shows the delay comparison of SEFAC and MUFAC, with SEFA as a benchmark. The plot shows that SEFAC and MUFAC have identical delay performance, and have an expected disadvantage as compared to SEFA at load 0.9 and above. This delay disadvantage is mainly the result of center-route limitation in the Clos-network switch architecture. Under light traffic loading, limited center-routes are more than the system's need; thus, the Clos-network switch architecture is transparent to FDL assignment. Therefore, SEFAC and MUFAC have compatible delay performance as compared to SEFA at light load. Under heavy traffic loading, center-route availabilities in the Clos-network switch architecture become a resource limitation; hence, SEFAC and MUFAC show delay disadvantages over SEFA at load 0.9 and above. On the other hand, as the offered load approaches 1, their difference becomes smaller. This may be due to the fact that the congestion mainly occurs at the FDL assignment in each switch module, rather than the route limitation through the CMs.

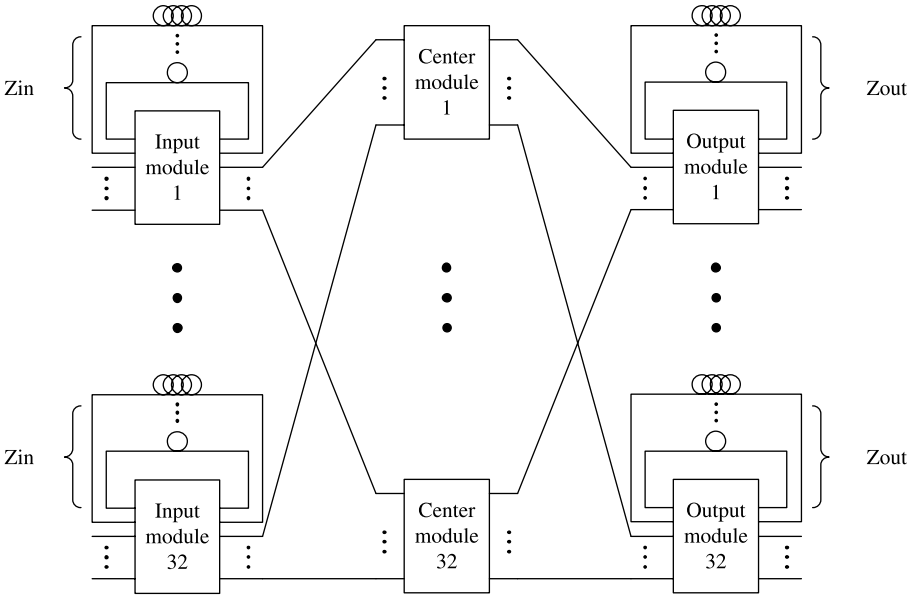


Figure 15.74 FDL distribution in three-stage OCNS.

15.6.5 Complexity Analysis of SEFAC and MUFAC

The time complexity of SEFAC is a function of the size of the three-stage SFI-OCNS. Since SEFAC has the similar operation to SEFA, SEFAC has the time complexity $K \times N \times (Q \times T)$, where K is the number of OMs, N is the number of output ports for each OM, Q is the number of nodes in the transition diagram G, and T is the time for each input request to search one node in the transition diagram G for output port and FDL availability. For instance, in a 1024×1024 SFI-OCNS, which has 32 IM, 32 CM, and 32 OM, each module has a size of 32×32, and each IM has 32 FDLs, then $K = 32$, $N = 32$. If we

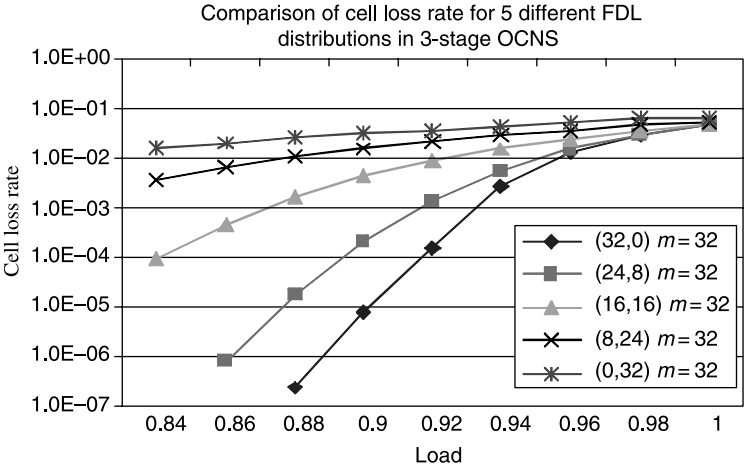


Figure 15.75 Performance comparison for five different FDL distributions in the OCNS.

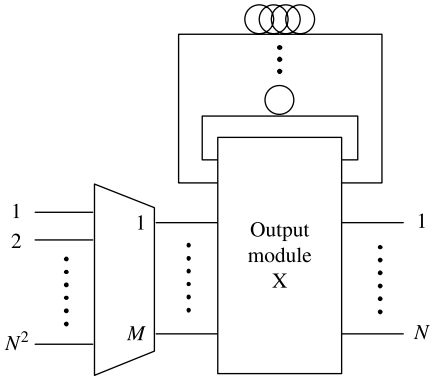


Figure 15.76 Knockout principle at OM of the OCNS.

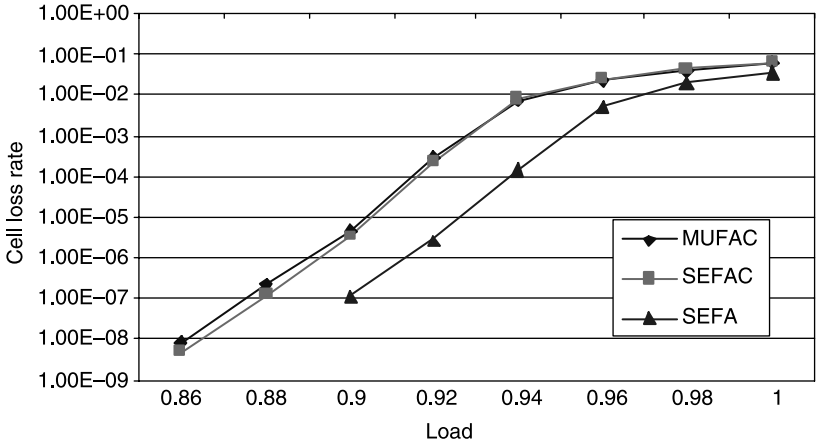


Figure 15.77 Loss comparison of SEFAC and MUFAC.

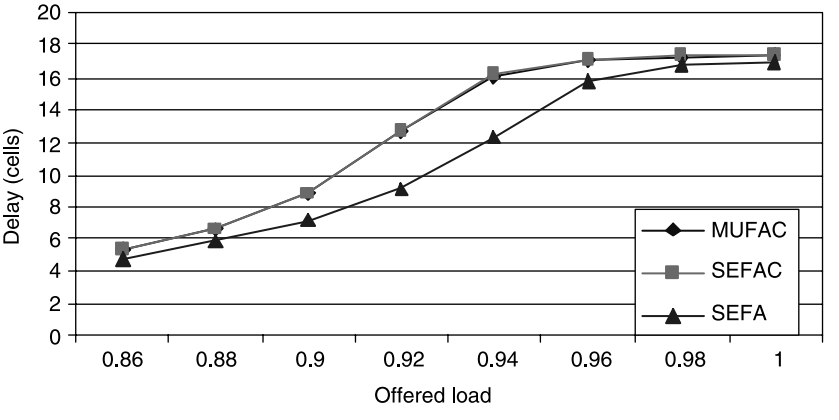


Figure 15.78 Delay performance comparison of SEFAC and MUFAC.

limit the maximum delay operation to 2, then Q, the total number of nodes in the transition diagram G, is 36. Let us assume $T = 10\,\text{ns}$. Thus, the total complexity of the SEFAC is $32 \times 32 \times (36 \times 10\,\text{ns}) = 369\,\mu\text{s}$.

To find the time complexity of MUFAC, let us consider the complexity of MUFAC at each cycle first. Suppose the time needed for a parent node to send out an unfulfilled request is T_r; the time needed for child nodes to make a grant decision is T_g, which includes a step of parallel AND operations to match the unfulfilled requests with the available output ports; the time needed to find available center-routes is T_c; the time needed for parent nodes to make accepting decisions is T_a, where T_a consists of $\log_2 F$ sequential steps of bit comparison (to grant the matches for each child node); and the time needed for all processing nodes to update information is T_u. Although requesting and updating are two different procedures in the MUFAC algorithm, these two tasks consist of only register accessing; so, they can be performed in parallel. Therefore, the time needed for these two tasks can be counted as one called $T_{r/u}$. Then the time for one cycle process is $T_g + T_a + T_{r/u}$. Moreover, let K be the number of cycles in each process, and let P be the number of nodes that act as parent nodes during the MUFAC process. The time complexity of MUFAC is $P \times K \times (T_g + T_a + T_{r/u})$. For example, a 1024×1024 OCNS that has 32 IMs, 32 CMs, and 32 OMs (each module has a size of 32×32), then $P = 1 + 7 = 8$ with limited delay operation of 2 and $K = 32$. Assume $T_g = T_{r/u} = 5\,\text{ns}$, then $T_a = (1 + \log_2 F) \times 5\,\text{ns} = 40\,\text{ns}$. Therefore, the total time complexity for MUFAC is $8 \times 32 \times 55\,\text{ns} = 14\,\mu\text{s}$.

REFERENCES

[1] P. E. Green, *Fiber Optic Communication Networks*. Prentice-Hall, Upper Saddle River, New Jercy, 1992.

[2] N. V. Srinivasan, "Add-drop multiplexers and cross-connects for multiwavelength optical networking," in *Proc. Tech. Dig., OFC'98*, San Jose, California, pp. 57–58 (Feb. 1998).

[3] C. K. Chan, F. Tong, L. K. Chen, and K. W. Cheung, "Demonstration of an add-drop network node with time slot access for high-speed WDMA dual bus/ring packet networks," in *Proc. Tech. Dig., OFC'98*, San Jose, California, pp. 62–64 (Feb. 1998).

[4] G. Chang, G. Ellinas, J. K. Gamelin, M. Z. Iqbal, and C. A. Brackett, "Multiwavelength reconfigurable WDM/ATM/SONET network testbed," *IEEE/OSA Journal of Lightwave Technology*, vol. 14, issue 6, pp. 1320–1340 (June 1996).

[5] R. E. Wanger, R. C. Alferness, A. A. M. Saleh, and M. S. Goodman, "MONET: Multiwavelength optical networking," *IEEE/OSA Journal of Lightwave Technology*, vol. 14, issue 6, pp. 1349–1355 (June 1996).

[6] S. Okamoto and K. Sato, "Optical path cross-connect systems for photonic transport networks," in *Proc. IEEE Global Telecommun. Conf.*, Houston, Texas, pp. 474–480 (Nov. 1993).

[7] A. K. Srivastava, J. L. Zyskind, Y. Sun, J. W. Sulhoff, C. Wolf, M. Zirngibl, R. Monnard, A. R. Charplyvy, A. A. Abramov, R. P. Espindola, T. A. Strasse, J. R. Pedrazzani, A. M. Vengsarkar, J. Zhou, and D. A. Ferrand, "1 Tb/s transmission of 100 WDM 10 Gb/s channels over 400 km of TrueWave^{TM} fiber," in *Postdeadline Papers, OFC'98*, San Jose, California, pp. PD10-1–PD10-4 (Feb. 1998).

[8] A. S. Tanenbaum, *Computer Networks*. Prentice-Hall, New Jersey, 1981.

[9] E. Arthurs, M. S. Goodman, H. Kobrinski, and M. P. Vecchi, "HYPASS: an optoelectronic hybrid packet switching system," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1500–1510 (Dec. 1988).

[10] T. T. Lee, M. S. Goodman, and E. Arthurs, "STAR-TRACK: a broadband optical multicast switch," *Bellcore Technical Memorandum Abstract*, 1989.

[11] A. Cisneros and C. A. Brackett, "A large ATM switch based on memory switches and optical star couplers," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 8, pp. 1348–1360 (Oct. 1991).

[12] M. J. Karol, M. G. Hluchyj, and S. P. Morgan, "Input versus output queueing on a space division packet switch," *IEEE Transactions on Communications*, vol. COM-35, no. 12, pp. 1347–1356 (Dec. 1987).

[13] E. Munter, L. Parker, and P. Kirkby, "A high-capacity ATM switch based on advanced electronic and optical technologies," *IEEE Communications Magazine*, vol. 33, issue 11, pp. 64–71 (Nov. 1995).

[14] Y. Nakahira, H. Inoue, and Y. Shiraishi, "Evaluation of photonic ATM switch architecture-proposal of a new switch architecture," in *Proc. International Switching Symposium*, Berlin, Germany, pp. 128–132 (1995).

[15] H. J. Chao and T.-S. Wang, "Design of an optical interconnection network for terabit IP router," in *Proc. IEEE LEOS'98*, Orlando, Florida, vol. 1, pp. 233–234 (Dec. 1998).

[16] D. Lesterlin, S. Artigaud, and H. Haisch, "Integrated laser/modulators for 10 Gbit/s system," in *Proc. 22nd European Conference on Optical Communication (ECOC'96)*, Olso, Norway, pp. 3.183–3.190 (Sep. 1996).

[17] M. G. Young, U. Koren, B. I. Milter, M. A. Newkirk, M. Chien, M. Zirngibl, C. Dragone, B. Tell, H. M. Presby, and G. Raybon, "A 16 × 1 wavelength division multiplexer with integrated distributed Bragg reflector laser and electroabsorption modulators," *IEEE Photonics Technology Letters*, vol. 5, no. 8, pp. 908–910 (Aug. 1993).

[18] K. C. Syao, K. Yang, X. Zhang, G. I. Haddad, and P. Bhattacharya, "16-channel monolithically integrated InP-based p-i-n/HBT photoreceiver array with 11-GHz channel bandwidth and low crosstalk," in *Proc. Optical Fiber Communication (OFC'97)*, Dallas, Texas, pp. 15–16 (Feb. 1997).

[19] I. Ogawa, F. Ebisawa, F. Hanawa, T. Hashimoto, M. Yanagisawa, K. Shuto, T. Ohyama, Y. Yamada, Y. Akahori, A. Himeno, K. Kato, N. Yoshimoto, and Y. Tohmon, "Lossless hybrid integrated 8-ch optical wavelength selector module using PLC platform and PLC-PLC direct attachment techniques," in *Proc. Optical Fiber Communication Conference* (*OFC'98*), San Jose, California, pp. 1–4 (Feb. 1998).

[20] O. Ishida, H. Takahashi, and Y. Inoue, "Digitally tunable optical filters using arrayed-waveguide grating (AWG) multiplexers and optical switches," *IEEE/OSA Journal of Lightwave Technology*, vol. 15, no. 2, pp. 321–327 (1997).

[21] K. Yamakoshi, K. Nakai, N. Matsuura, E. Oki, R. Kawano, and N. Yamanaka, "5-Tbit/s frame-based ATM switching system using 2.5-Gbit/s/spl times/8 optical WDM," in *Proc. IEEE International Conference on Communications*, Helsinki, Finland, vol. 10, pp. 3117–3121 (June 2001).

[22] C. K. Chan, K. L. Sherman, and M. Zirngibl, "A fast 100-channel wavelength-tunable transmitter for optical packet switching," *IEEE Photonics Technology Letters*, vol. 13, issue 7, pp. 729–731 (July 2001).

[23] K. R. Tamura, Y. Inoue, K. Sato, T. Komukai, A. Sugita, and M. Nakazawa, "A discretely tunable mode-locked laser with 32 wavelengths and 100-GHz channel spacing using an arrayed waveguide grating," *IEEE Photonics Technology Letters*, vol. 13, issue 11, pp. 1227–1229 (Nov. 2001).

[24] S. Nakamura, Y. Ueno, and K. Tajima, "168-Gb/s all-optical wavelength conversion with a symmetric-Mach-Zehnder-type switch," *IEEE Photonics Technology Letters*, vol. 13, issue 10, pp. 1091–1093 (Oct. 2001).

[25] J. Leuthold, B. Mikkelsen, G. Raybon, C. H. Joyner, J. L. Pleumeekers, B. I. Miller, K. Dreyer, and R. Behringer, "All-optical wavelength conversion between 10 and 100 Gb/s with SOA delayed-interference configuration," *Optical and Quantum Electronic*, vol. 33, no. 7–10, pp. 939–952 (2001).

[26] H. Takara, S. Kawanishi, Y. Yamabayashi, Y. K. Tohmori, K. Takiguchi, Y. K. Magari, I. Ogawa, and A. Himeno, "Integrated optical-time division multiplexer based on planar lightwave circuit," *Electronics Letters*, vol. 35, issue 15, pp. 1263–1264 (1999).

[27] D. T. K. Tong, K.-L. Deng, B. Mikkelsen, G. Ranbon, K. F. Dreyer, and J. E. Johnson, "160 Gbit/s clock recovery using electroabsorption modulator-based phase-locked loop," *Electronics Letters*, vol. 36, no. 23, pp. 1951–1952 (2000).

[28] K.-L. Deng, D. T. K. Tong, C.-K. Chan, K. F. Dreyer, and J. E. Johnson, "Rapidly reconfigurable optical channel selector using RF digital phase shifter for ultra-fast OTDM networks," *Electronics Letters*, vol. 36, no. 20, pp. 1724–1725 (2000).

[29] Z. Hass, "The "staggering switch": an electrically controlled optical packet switch," *IEEE Journal of Lightwave Technology*, vol. 11, no. 5, pp. 925–936 (May 1993).

[30] D. Chiaroni, C. Chauzat, D. D. Bouard, and M. Sotom, "Sizeability analysis of a high-speed photonic packet switching architecture," in *Proc. 21st Eur. Conf. on Opt. Comm.* (*ECOC'95*), Brussels, Belgium, pp. 793–796 (Sep. 1995).

[31] G. H. Duan, J. R. Fernandez, and J. Garabal, "Analysis of ATM wavelength routing systems by exploring their similitude with space division switching," in *Proc. IEEE International Conference on Communication*, Dallas, Texas, vol. 3, pp. 1783–1787 (June 1996).

[32] Y. Chai, J. H. Chen, F. S. Choa, J. P. Zhang, J. Y. Fan, and W. Lin, "Scalable and modularized optical random access memories for optical packet switching networks," in *Proc. CLEO'98*, San Francisco, California, pp. 397 (May 1998).

[33] Y. Chai, J. H. Chen, X. J. Zhao, J. P. Zhang, J. Y. Fan, F. S. Choa, and W. Lin, "Optical DRAMs using refreshable WDM loop memories," in *Proc. ECOC'98*, Madrid, Spain, pp. 171–172 (Sep. 1998).

[34] R. Langenhorst, M. Eiselt, W. Pieper, G. Groossleupt, R. Ludwig, L. Kuller, "Fiber loop optical buffer," *IEEE Journal of Lightwave Technology*, vol. 14, no. 3, pp. 324–335 (1996).

[35] G. Bendelli, M. Burzio, M. Calzavara, P. Cinato, P. Gambini, M. Puleo, E. Vezzoni, F. Delorme, and H. Nakajima, "Photonic ATM switch based on a multiwavelength fiber-loop buffer," in *Proc. OFC'95*, San Diego, California, pp. 141–142 (Feb. 1995).

[36] Y. Yamada, K. Sasayama, and K. Habara, "Transparent optical-loop memory for optical FDM packet buffering with differential receiver," in *Proc. ECOC'96*, Olsa, Norway, pp. 317–320 (Sept. 1996).

[37] J. Ramamirtham and J. Turner, "Time sliced optical burst switching," in *Proc. IEEE INFOCOM'03*, San Francisco, California, pp. 2030–2038 (Apr. 2003).

[38] M. C. Chia, D. K. Hunter, I. Andonovic, P. Ball, I. Wright, S. P. Ferguson, K. M. Guild, and M. J. O. Mahony, "Packet loss and delay performance of feedback and feed-forward arrayed-waveguide gratings-based optical packet switches with WDM inputs–outputs," *IEEE/OSA Journal of Lightwave Technology*, vol. 19, no. 9, pp. 1241–1254 (Sept. 2001).

[39] C. Qiao and M. Yoo, "Optical burst switching (OBS) – A new paradigm for an optical Internet," *Journal of High Speed Networks*, vol. 8, no. 1, pp. 69–84 (1999).

[40] I. Baldine, G. N. Rouskas, H. G. Perros, and D. Stevenson, "Jumpstart: a just-in-time signaling architecture for WDM burst-switched networks," *IEEE Communications Magazine*, vol. 40, no. 2, pp. 82–89 (Feb. 2002).

[41] M. J. Karol, "Shared-memory optical packet (ATM) switch," *Multigigabit Fiber Communications Systems*, vol. 2024, pp. 212–222 (July 1993).

[42] S. Y. Liew, G. Hu, and H. J. Chao, "Scheduling algorithms for shared fiber-delay-line optical packet switches. Part I: The single-stage case," *IEEE Journal of Lightwave Technology*, vol. 23, issue 4, pp. 1586–1600 (Apr. 2005).

[43] N. Huang, G. Liaw, and C. Wang, "A novel all-optical transport network with time-shared wavelength channels," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, pp. 1863–1875 (Oct. 2000).

[44] B. Wen and K. M. Sivalingam, "Routing, wavelength and time-slot assignment in time division multiplexed wavelength-routed optical WDM networks," in *Proc. IEEE INFOCOM'02*, New York, New York, vol. 3, pp. 1442–1450 (Apr. 2002).

[45] R. Srinivasan and A. K. Somani, "A generalized framework for analyzing time-space switched optical networks," in *Proc. IEEE INFOCOM'01*, Anchorage, Alaska, pp. 179–188 (Apr. 2001).

[46] S. Jiang, G. Hu, S. Y. Liew, and H. J. Chao, "Scheduling algorithms for shared fiber-delay-line optical packet switches, Part II: The 3-stage Clos-Network case," *IEEE Journal of Lightwave Technology*, vol. 23, issue 4, pp. 1601–1609 (Apr. 2005).

[47] C. Clos, "A study of non-blocking switching networks," *Bell System Technical Journal*, vol. 32, no. 3, pp. 406–424 (Mar. 1953).

[48] Y. S. Yeh, M. G. Hluchyj, and A. S. Acampora, "The knockout switch: a simple, modular architecture for high-performance switching," *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 8, pp. 1274–1283 (Oct. 1987).