CHAPTER 1

INTRODUCTION

The scalable and distributed nature of the Internet continuously contributes to a wild and rapid growth of its population, including the number of users, hosts, links, and emerging applications. The great success of the Internet thus leads to exponential increases in traffic volumes, stimulating an unprecedented demand for the capacity of the core network.

Network providers therefore face the need of providing a new network infrastructure that can support the growth of traffic in the core network. Advances in fiber throughput and optical transmission technologies have enabled operators to deploy capacity in a dramatic fashion. However, the advancement in packet switch/router technologies is rather slow, so that it is still not able to keep pace with the increase in link transmission speed.

Dense-wavelength-division-multiplexing (DWDM) equipment is installed on each end of the optical fiber to multiplex wavelengths (i.e., channels) over a single fiber. For example, a 128-channel OC-192 (10 Gbit/s) DWDM system can multiplex the signals to achieve a total capacity of 1.2 Tbit/s. Several vendors are expected to enter trials for wide area DWDM networks that support OC-768 (40 Gbit/s) for each channel in the near future.

Another advanced optical technology that is being deployed in the optical network is the optical cross connect (OXC) system. Since the optical-to-electrical-to-optical conversions do not occur within the system, transmission interfaces are transparent. The OXC System is based on the microelectromechanical systems (MEMS) technology, where an array of hundreds or thousands of electrically configurable microscopic mirrors is fabricated on a

single substrate to direct light. The switching scheme is based on freely moving mirrors being rotated around micromachined hinges with submillisecond switching speed. It is rate- and format-independent.

As carriers deploy fiber and DWDM equipment to expand capacity, terabit packet switching technologies are required to aggregate high-bit-rate links while achieving higher utilization on the links. Although OXC systems have high-speed interfaces (e.g., 10 or 40 Gbit/s) and large switching capacity (e.g., 10–40 Tbit/s), the granularity of the switching is coarse, e.g., 10 or 40 Gbit/s. As a result, it is required to have high-speed and large-capacity packet switches/routers to aggregate lower-bit-rate traffic to 10 or 40 Gbit/s links. The aggregated traffic can be delivered to destinations through DWDM transmission equipment or OXC systems. The terabit packet switches that are critical elements of the Internet network infrastructure must have switch fabric capable of supporting terabit speeds to eliminate the network bottlenecks. Core terabit switches/routers must also deliver low latency and guaranteed delay variance to support real-time traffic. As a result, quality-of-service (QoS) control techniques, such as traffic shaping, packet scheduling, and buffer management, need to be incorporated into the switches/routers.

Asynchronous transfer mode (ATM) is revolutionizing the telecommunications infrastructure by transmitting integrated voice, data, and video at very high speed. The current backbone network mainly consists of ATM switches and IP routers, where ATM cells and IP packets are carried on an optical physical layer such as the Synchronous Optical Network (SONET). ATM also provides different QoS requirements for various multimedia services. Readers who are interested in knowing the SONET frame structure, the ATM cell format, and the functions associated with SONET/ATM layers are referred to the Appendix.

Along with the growth of the Internet, IP has become the dominant protocol for data traffic and is making inroads into voice transmission as well. Network providers recognize the cost savings and performance advantages of converging voice, data, and video services onto a common network infrastructure, instead of an overlayered structure. Multi-protocol label switching (MPLS) is a new technology combining the advantageous features of the ATM network, short labels and explicit routing, and the connectionless datagram of the IP network. The MPLS network also provides traffic engineering capability to achieve bandwidth provisioning, fast restoration, load balancing, and virtual private network (VPN) services. The so-called label switching routers (LSRs) that route packets can be either IP routers, ATM switches, or frame relay switches. In this book, we will address the issues and technologies of building a scalable switch/router with large capacity, e.g., several terabits per second.

In the rest of this chapter, we briefly describe the ATM network, ATM switch systems, IP router systems, and switch design criteria and performance requirements.

1.1 ATM SWITCH SYSTEMS

1.1.1 Basics of ATM Networks

ATM protocol corresponds to layer 2 as defined in the open systems interconnection (OSI) reference model. ATM is connection-oriented. That is, an end-to-end connection (or *virtual channel*) needs to be set up before routing ATM cells. Cells are routed based on two important values contained in the 5-byte cell header: the *virtual path identifier* (VPI) and *virtual channel identifier* (VCI), where a virtual path consists of a number of virtual channels. The number of bits allocated for a VPI depends on the type of interface. If it is the *user network interface* (UNI), between the user and the first ATM switch, 8 bits are provided for the VPI. This means that up to $2^8 = 256$ virtual paths are available at the user access point. On the other hand, if the it is the *network node interface* (NNI), between the intermediate ATM switches, 12 bits are provided for the VPI. This indicates that there are $2^{12} = 4096$ possible virtual paths between ATM switches. In both UNI and NNI, there are 16 bits for the VCI. Thus, there are $2^{16} = 65{,}536$ virtual channels for each virtual path.

The combination of the VPI and the VCI determines a specific virtual connection between two ends. Instead of having the same VPI/VCI for the whole routing path, the VPI/VCI is determined on a per-link basis and changes at each ATM switch. Specifically, at each incoming link to a switch node, a VPI/VCI may be replaced with another VPI/VCI at the output link with reference to a table called a *routing information table* (RIT) in the ATM switch. This substantially increases the possible number of routing paths in the ATM network.

The operation of routing cells is as follows. Each ATM switch has its own RIT containing at least the following fields: old VPI/VCI, new VPI/VCI, *output port address*, and *priority* field (optional). When an ATM cell arrives at an input line of the switch, it is split into the 5-byte header and the 48-byte payload. By using the VPI/VCI contained in the header as the old VPI/VCI value, the switch looks in the RIT for the arriving cell's new VPI/VCI. Once the match is found, the old VPI/VCI value is replaced with the new VPI/VCI value. Moreover, the corresponding output port address and priority field are attached to the 48-byte payload of the cell, before it is sent to the switch fabric. The output port address indicates to which output port the cell should be routed. There are three modes of routing operations within the switch fabric: the *unicast mode* refers to the mode in which a cell is routed to a specific output port, the *multicast mode* refers to the mode in which a cell is routed to a number of output ports, and the *broadcast mode* refers to the mode in which a cell is routed to all output ports. In the unicast mode, $\log_2 N$ bits, where N is the number of input/output ports, are sufficient to indicate any possible output port. However, in the multicast/broadcast modes, N bits, each associated with a particular output

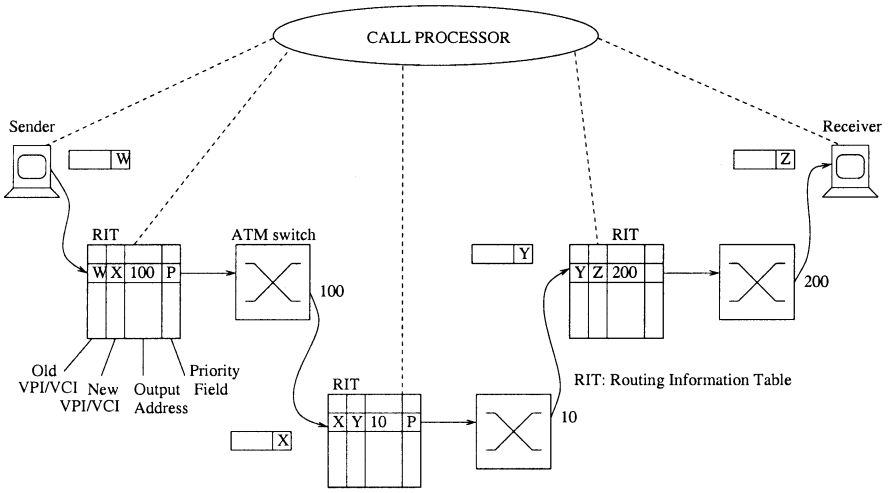


Fig. 1.1 VPI/VCI translation along the path.

port, are needed in a single-stage switch. The priority field enables the switch to selectively transmit cells to the output ports or discard them when the buffer is full, according to service requirements.

ATM connections either are preestablished through provisioning or are set up dynamically on demand using signaling, such as UNI signaling and private network–network interface (PNNI) routing signaling. The former is referred to permanent virtual connections (PVCs), while the latter is referred to switched virtual connections (SVCs). For SVCs, the RIT is updated by a *call processor* during the call setup, which finds an appropriate routing path between the source and the destination. The VPI/VCI of every link along the path, the output port addresses of the switches, and the priority field are determined and filled into the table by the call processor. The call processor has to ensure that at each switch, the VPI/VCI of the cells coming from different connections but going to the same output port are different. In practice, there is a call processor for every ATM switch. For simplicity, Figure 1.1 just shows a call processor to update the RIT of each switch in a conceptual way.

With respect to Figure 1.1, once a call setup is completed, the source starts to send a cell whose VPI/VCI is represented by W. As soon as this cell arrives at the first ATM switch, the entries of the table are searched. The matched entry is found with a new VPI/VCI X, which replaces the old VPI/VCI W. The corresponding output port address (whose value is 100) and the priority field are attached to the cell so that the cell can be routed to output port 100 of the first switch. At the second ATM switch, the VPI/VCI of the cell whose value is X is updated with a new value Y. Based on the output port address obtained from the table, the incoming cell is routed to

output port 10. This operation repeats in other switches along the path to the destination. Once the connection is terminated, the call processor deletes the associated entries of the routing tables along the path.

In the multicast case, a cell is replicated into multiple copies and each copy is routed to an output port. Since the VPI/VCI of each copy at the output port can be different, VPI/VCI replacement usually takes place at the output instead of the input. As a result, the routing table is usually split into two parts, one at the input and the other at the output. The former has two fields in the RIT: the old VPI/VCI and the N-bit routing information. The latter has three fields in the RIT: the input port number, the old VPI/VCI, and the new VPI/VCI. The combination of the input port number and the old VPI/VCI can uniquely identify the multicast connection and is used as an index to locate the new VPI/VCI at the output. Since multiple VPI/VCIs from different input ports can merge to the same output port and have the identical old VPI/VCI value, it thus has to use extra information as part of the index for the RIT. Using the input port number is a natural and easy way.

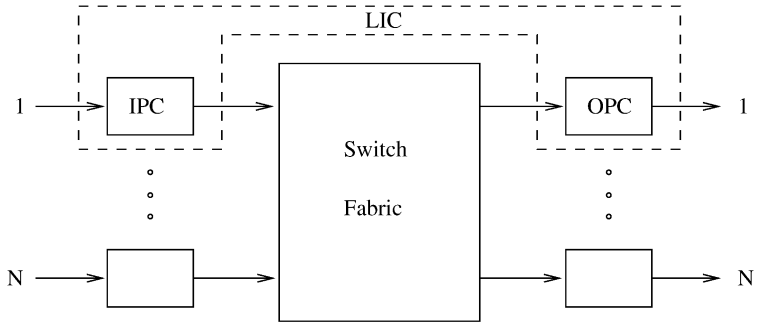
1.1.2 ATM Switch Structure

Figure 1.2(a) depicts a typical ATM switch system model, consisting of input port controllers (IPCs), a switch fabric, and output port controllers (OPCs). In practice, the IPC and the OPC are usually built on the same printed circuit board, called the line interface card (LIC). Multiple IPCs and OPCs can be built on the same LIC. The center switch fabric provides interconnections between the IPCs and the OPCs. Figure 1.2(b) shows a chasis containing a power supply card, a CPU card to perform operation, administration, and maintenance (OAM) functions for the switch system, a switch fabric card, and multiple LICs. Each LIC has a transmitter (XMT) and a receiver (RCV).

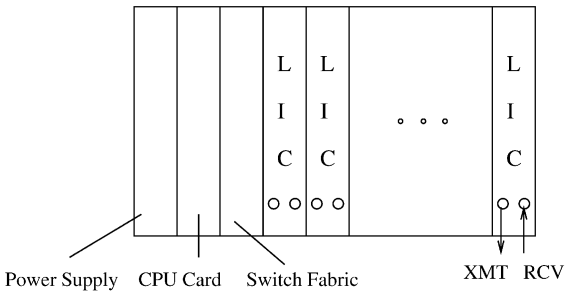
As shown in Figure 1.3(a), each IPC terminates an incoming line and extracts cell headers for processing. In this example, optical signals are first converted to electrical signals by an optical-to-electrical (O/E) converter and then terminated by an SONET framer. Cell payloads are stored in a first-in, first-out (FIFO) buffer, while headers are extracted for routing processing. Incoming cells are aligned before being routed in the switch fabric, which greatly simplifies the design of the switch fabric. The cell stream is slotted, and the time required to transmit a cell across to the network is a time slot.

In Figure 1.3(b), cells coming from the switch fabric are stored in a FIFO buffer.[1] Routing information (and other information such as a priority level,

[1]For simplicity, here we use a FIFO for the buffer. To meet different QoS requirements for different connections, some kind of scheduling policies and buffer management schemes may be applied to the cells waiting to be transmitted to the output link. As a result, the buffer may not be as simple as a FIFO.



(a)



(b)

Fig. 1.2 An ATM switch system example.

if any) will be stripped off before cells are written to the FIFO. Cells are then carried in the payload of SONET frames, which are then converted to optical signals through an electrical-to-optical (E/O) converter.

The OPC can transmit at most one cell to the transmission link in each time slot. Because cells arrive randomly in the ATM network, it is likely that more than one cell is destined for the same output port. This event is called output port contention (or conflict). One cell will be accepted for transmission, and the others must be either discarded or buffered. The location of the buffers not only affects the switch performance significantly, but also affects the switch implementation complexity. The choice of the contention resolution techniques is also influenced by the location of the buffers.

There are two methods of routing cells through an ATM switch fabric: *self-routing* and *label routing*. In self-routing, an output port address field (A) is prepended to each cell at the input port before the cell enters to the switch

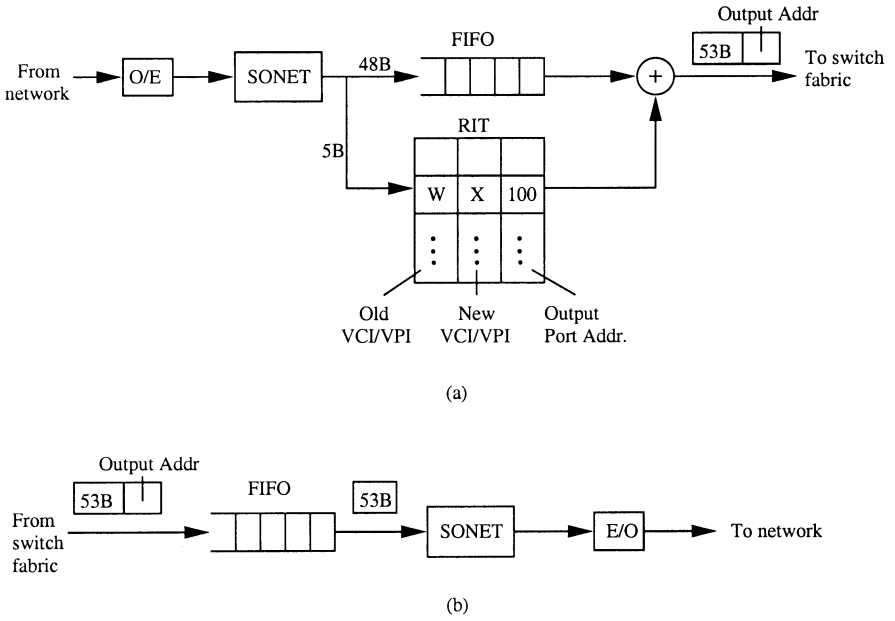


Fig. 1.3 Input and output port controller block diagram.

fabric. This field, which has $\log_2 N$ bits for unicast cells or N bits for multicast/broadcast cells, is used to navigate the cells to their destination output ports. Each bit of the output port address field is examined by each stage of the switch element. If the bit is 0, the cell is routed to the upper output of the switch element. If the bit is 1, it is routed to its lower output. As shown in Figure 1.4, a cell whose output port address is 5 (101) is routed to input port 2. The first bit of the output port address (1) is examined by the first stage of the switch element. The cell is routed to the lower output and goes to the second stage. The next bit (0) is examined by the second stage, and the cell is routed to the upper output of the switch element. At the last stage of the switch element, the last bit (1) is examined and the cell is routed to its lower output, corresponding to output port 5. Once the cell arrives at the output port, the output port address is removed.

In contrast, in label routing the VPI/VCI field within the header is used by each switch module to make the output link decision. That is, each switch module has a VPI/VCI lookup table and switches cells to an output link according to the mapping between VPI/VCI and input/output links in the table. Label routing does not depend on the regular interconnection of switching elements as self-routing does, and can be used arbitrarily wherever switch modules are interconnected.

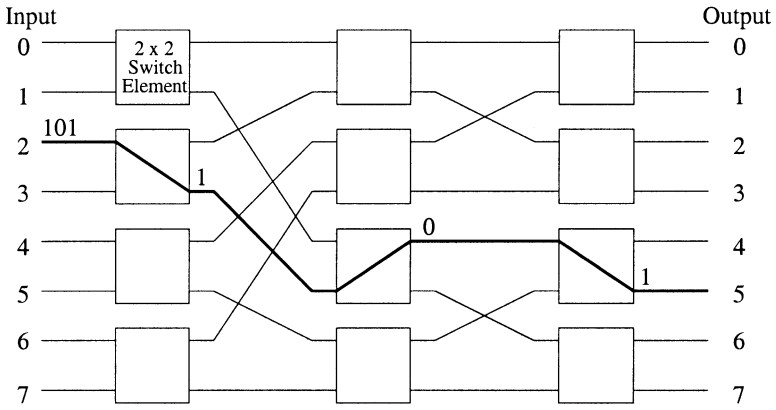


Fig. 1.4 An example of self-routing in a delta network.

1.2 IP ROUTER SYSTEMS

The Internet protocol corresponds to layer 3 as defined in the OSI reference model. The Internet, as originally conceived, offers best-effort data delivery. In contrast to ATM, which is a connection-oriented protocol, IP is a connectionless protocol. A user sends IP packets to IP networks without setting up any connection. As soon as a packet arrives at an IP router, the router decides to which output link the packet is to be routed, based on its IP address in the packet overhead, and transmits it to the output link.

To meet the demand for QoS for IP networks, the Internet Engineering Task Force (IETF) has proposed many service models and mechanisms, including the integrated services/resource reservation protocol (RSVP) model, the differentiated services (DS) model, MPLS, traffic engineering, and constraint-based routing [2]. These models will enhance today's IP networks to support a variety of service applications.

1.2.1 Functions of IP Routers

IP routers' functions can be classified into two categories, datapath functions and control functions [1].

The datapath functions are performed on every datagram that passes through the router. These include the forwarding decision, switching through the backplane, and output link scheduling. When a packet arrives at the forwarding engine, its destination IP address is first masked by the subnet mask (logical AND operation), and the resulted address is used to look up the forwarding table. A so-called longest-prefix matching method is used to find the output port. In some application, packets are classified based on the combined information of IP source/destination addresses, transport layer

port numbers (source and destination), and type of protocol: total 104 bits. Based on the result of classification, packets may be either discarded (firewall application) or handled at different priority levels. Then, the time-to-live (TTL) value is decremented and a new header checksum is calculated. Once the packet header is used to find an output port, the packet is delivered to the output port through a switch fabric. Because of contention by multiple packets destined for the same output port, packets are scheduled to be delivered to the output port in a fair manner or according to their priority levels.

The control functions include the system configuration, management, and exchange of routing table information. These are performed relatively infrequently. The route controller exchanges the topology information with other routers and constructs a routing table based on a routing protocol (e.g., RIP and OSPF). It can also create a forwarding table for the forwarding engine. Since the control function is not processed for each arriving packet, it does not have a speed constraint and is implemented in software.

1.2.2 Architectures of IP Routers

1.2.2.1 Low-End Routers In the architecture of the earliest routers, the forwarding decision and switching function were implemented in a central CPU with a shared central bus and memory, as shown in Figure 1.5. These functions are performed based on software. Since this software-based structure is cost-effective, it is mainly used by low-end routers. Although CPU performance has improved with time, it is still a bottleneck to handle all the packets transmitted through the router with one CPU.

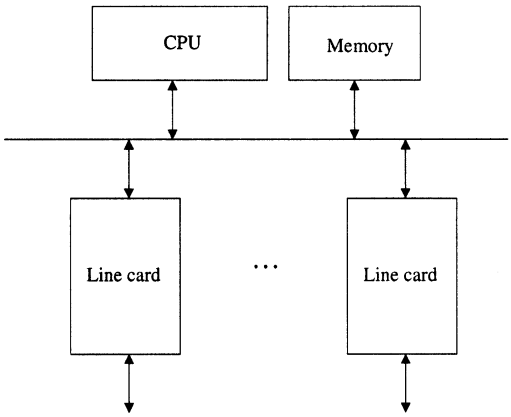


Fig. 1.5 Low-end router structure.

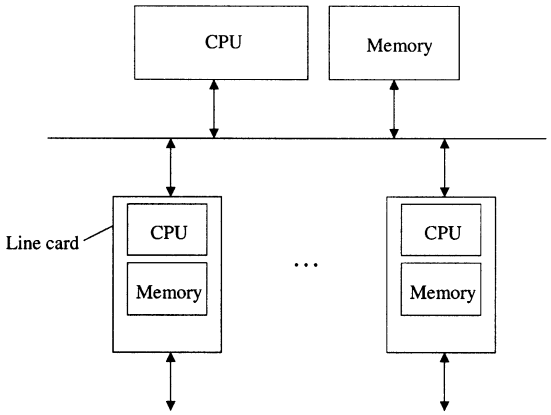


Fig. 1.6 Medium-size router structure.

1.2.2.2 Middle-Size Routers To overcome the limitation of one central CPU, medium-size routers use router structure where each line card has a CPU and memory for packet forwarding, as shown in Figure 1.6. This structure reduces the central CPU load, because incoming packets are processed in parallel by the CPU associated with each line card, before they are sent through the shared bus to the central CPU and finally to the memory of the destined output line card. However, when the number of ports and the port speed increase, this structure still has a bottleneck at the central CPU and shared bus.

1.2.2.3 High-End Routers In current high-performance routers, the forwarding engine is implemented in each associated ingress line card, and the switching function is implemented by using switch-fabric boards. As shown in Figure 1.7, a high-performance router consists of a route controller, forwarding engine, switch fabric, and output port scheduler. In this structure, multiple line cards can communicate with each other. Therefore, the router throughput is improved.

Once a packet is switched from an ingress line card to an egress line card, it is usually buffered at the output port that is implemented in the egress line card. This is because multiple packets may be destined for the same output port at the same time. Only one packet can be transmitted to the network at any time, and the rest of them must wait at the output buffer for the next transmission round. In order to provide differentiated service for different flows, it is necessary to schedule packets according to their priority levels or the allocated bandwidth. There may also be some buffer management, such as random early detection (RED), to selectively discard packets to achieve certain goals (e.g., desynchronizing the TCP flows). How to deliver packets from the forward engines to the output ports through the switch fabric is one of main topics to be discussed in this book.

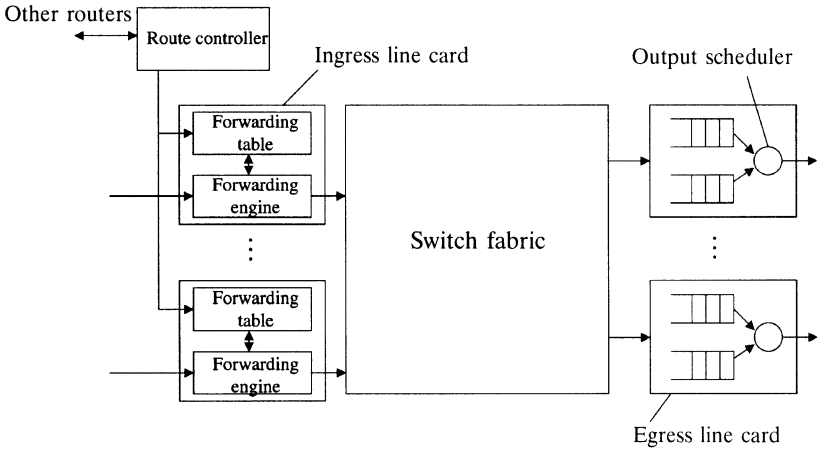


Fig. 1.7 High-end router structure.

For high-performance routers, datapath functions are most often implemented in hardware. If any datapath function cannot be completed in the interval of the minimum packet length, the router will not be able to operate at so-called wire speed, nor accommodate all arriving shortest packets at the same time.

Packets can be transferred across the switch fabric in small fixed-size data units, or as variable-length packets. The fixed-size data units are called cells (they do not have to be the same length as in ATM, 53 bytes). For the former case, each variable-length packet is segmented into cells at the input ports of the switch fabric and reassembled to that packet at the output ports. To design a high-capacity switch fabric, it is highly desirable to transfer packets in cells, for the following reasons. Due to output port contention, there is usually an arbiter to resolve contention among the input packets (except for output-buffered switches, since all packets can arrive at the same output port simultaneously—though, due to the memory speed constraint, the switch capacity is limited). For the case of delivering cells across the switch fabric, scheduling is based on the cell time slot. At the beginning of a time slot, the arbiter will determine which inputs' cells can be sent to the switch fabric. The arbitration process is usually overlapped (or pipelined) with the cell transmission. That is, while cells that are arbitrated to be winners in last time slot are being sent through the switch fabric in the current time slot, the arbiter is scheduling cells for the next time slot. From the point of view of hardware implementation, it is also easier if the decision making, data transferring, and other functions are controlled in a synchronized manner, with cell time slots.

For the case of variable-length packets, there are two alternatives. In the first, as soon as an output port completely receives a packet from an input port, it can immediately receive another packet from any input ports that

have packets destined for it. The second alternative is to transfer the packet in a cell-based fashion with the constraint that cells that belong to the same packet will be transmitted to the output port consecutively, and not interleaved with other cells. The differences between these two options have to do with event-driven vs. slot-driven arbitration. In the case of variable-length packets, the throughput is degraded, especially when supporting multicast services. Although the implementations of both alternatives are different, the basic cell transmission mechanisms are the same. Therefore, we describe an example of the variable-length-based operation by considering the first alternative for simplicity.

The first alternative starts the arbitration cycle as soon as a packet is completely transferred to the output port. It is difficult to have parallelism for arbitration and transmission, due to the lack of knowledge of when the arbitration can start. Furthermore, when a packet is destined for multiple output ports by taking advantage of the replication capability of the switch fabric (e.g., crossbar switch), the packet has to wait until all desired output ports become available. Consider a case where a packet is sent to output ports x and y. When output x is busy in accepting a packet from another input and output y is idle, the packet will not be able to send to output y until port x becomes available. As a result, the throughput for port y is degraded. However, if the packet is sent to port y without waiting for port x becomes available, as soon as port x becomes available, the packet will not be able to send to port x, since the remaining part of the packet is being sent to port y. One solution is to enable each input port to send multiple streams, which will increase the implementation complexity. However, if cell switching is used, the throughput degradation is only a cell slot time, as oppose to, when packet switching is used, the whole packet length, which can be a few tens or hundreds of cell slots.

1.2.2.4 Switch Fabric for High-End IP Routers Although most of today's low-end to medium-size routers do not use switch fabric boards, high-end backbone IP routers will have to use the switch fabric as described in Section 1.2.2.3 in order to achieve the desired capacity and speed as the Internet traffic grows. In addition, since fixed-size cell-switching schemes achieve higher throughput and simpler hardware design, we will focus on the switch architecture using cell switching rather than packet switching.

Therefore, the high-speed switching technologies described in this book are common to both ATM switches and IP routers. The terms of packet switches and ATM switches are interchangeable. All the switch architectures discussed in this book deal with fixed-length data units. Variable-length packets in IP routers, Ethernet switches, and frame relay switches, are usually segmented into fixed-length data units (not necessarily 53 bytes like ATM cells) at the inputs. These data units are routed through a switch fabric and reassembled back to the original packets at the outputs.

Differences between ATM switches and IP routers systems lie in their line cards. Therefore, both ATM switch systems and IP routers can be constructed by using a common switch fabric with appropriate line cards.

1.3 DESIGN CRITERIA AND PERFORMANCE REQUIREMENTS

Several design criteria need to be considered when designing a packet switch architecture. First, the switch should provide bounded delay and small cell loss probability while achieving a maximum throughput close to 100%. Capability of supporting high-speed input lines is also an important criterion for multimedia services, such as video conferencing and videophone. Self-routing and distributed control are essential to implement large-scale switches. Serving packets based on first come, first served provides correct packet sequence at the output ports. Packets from the same connection need to be served in sequence without causing out of order.

Bellcore has recommended performance requirements and objectives for broadband switching systems (BSSs) [3]. As shown in Table 1.1, three QoS classes and their associated performance objectives are defined: *QoS class* 1, *QoS class* 3, and *QoS class* 4. QoS class 1 is intended for stringent cell loss applications, including the circuit emulation of high-capacity facilities such as DS3. It corresponds to service class A, defined by ITU-T study group XIII. QoS class 3 is intended for low-latency, connection-oriented data transfer applications, corresponding to service class C in ITU-T study group XIII. QoS Class 4 is intended for low-latency, connectionless data transfer application, corresponding to service class D in ITU-T study group XIII.

The performance parameters used to define QoS classes 1, 3, and 4 are cell loss ratio, cell transfer delay, and two-point cell delay variation (CDV). The values of the performance objectives corresponding to a QoS class depend on the status of the cell loss priority (CLP) bit (CLP = 0 for high

TABLE 1.1 Performance Objective across BSS for ATM Connections Delivering Cells to an STS-3c or STS-12c Interface

Performance Parameter	CLP	QoS 1	QoS 3	QoS 4
Cell loss ratio	0	$< 10^{-10}$	$< 10^{-7}$	$< 10^{-7}$
Cell loss ratio	1	N/S[a]	N/S	N/S
Cell transfer delay (99th percentile)[b]	1/0	150 μs	150 μs	150 μs
Cell delay variation (10^{-10} quantile)	1/0	250 μs	N/S	N/S
Cell delay variation (10^{-7} quantile)	1/0	N/S	250 μs	250 μs

[a] N/S not specified.
[b] Includes nonqueuing related delays, excluding propagation. Does not include delays due to processing above ATM layer.

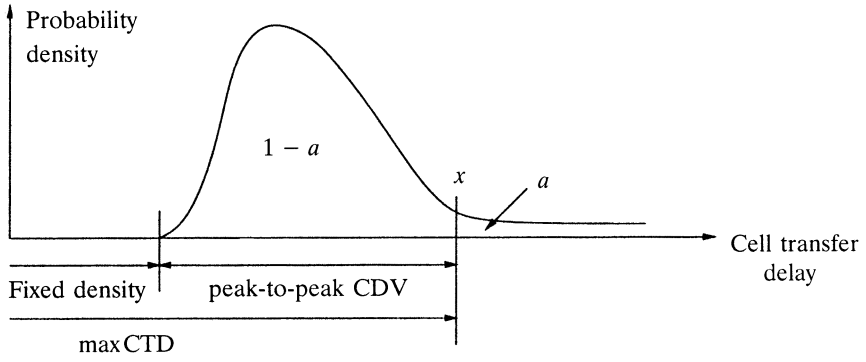


Fig. 1.8 Distribution of cell transfer delay.

priority, and CLP = 1 for low priority), which is initially set by the user and can be changed by a BSS within the connection's path.

Figure 1.8 shows a typical distribution of the cell transfer delay through a switch node. The fixed delay is attributed to the delay of table lookup and other cell header processing, such as header error control (HEC) byte examination and generation. For QoS classes 1, 3, and 4, the probability of cell transfer delay (CTD) greater than 150 μs is guaranteed to be less than $1 - 0.99$, that is, Prob[CTD $>$ 150 μs] $< 1 - 99\%$. For this requirement, $a = 1\%$ and $x = 150$ μs in Figure 1.8. The probability of CDV greater than 250 μs is required to be less than 10^{-10} for QoS class 1, that is, Prob[CDV $>$ 250 μs] $< 10^{-10}$.

REFERENCES

1. N. McKeown, "A fast switched backplane for a gigabit switched router," *Business Commun. Rev.*, vol. 27, no. 12, Dec. 1997.
2. X. Xiao and L. M. Ni, "Internet QoS: a big picture," IEEE Network, pp. 8–18, March/April 1999.
3. Bellcore, "Broadband switching system (BSS) generic requirements, BSS performance," GR-110-CORE, Issue 1, Sep. 1994.