CHAPTER 1

INTRODUCTION

The Internet, with its robust and reliable Internet Protocol (IP), is widely considered the most reachable platform for the current and next generation information infrastructure. The virtually unlimited bandwidth of optical fiber has tremendously increased the data transmission speed over the past decade. Availability of unlimited bandwidth has stimulated high-demand multimedia services such as distance learning, music and video download, and videoconferencing. Current broadband access technologies, such as digital subscriber lines (DSLs) and cable television (CATV), are providing affordable broadband connection solutions to the Internet from home. Furthermore, with Gigabit Ethernet access over dark fiber to the enterprise on its way, access speeds are expected to largely increase. It is clear that the deployment of these broadband access technologies will result in a high demand for large Internet bandwidth. To keep pace with the Internet traffic growth, researchers are continually exploring faster transmission and switching technologies. The advent of optical transmission technologies, such as dense wave division multiplexing (DWDM), optical add-drop multiplexers, and ultra-long-haul lasers have had a large influence on lowering the costs of digital transmission. For instance, 300 channels of 11.6 Gbps can be wavelength-division multiplexed on a single fiber and transmitted over 7000 km [1]. In addition, a 1296 × 1296 optical cross-connect (OXC) switching system using micro-electro-mechanical systems (MEMS) with a total switching capacity of 2.07 petabits/s has been demonstrated [2]. In the rest of this chapter, we explore state-of-the-art network infrastructure, future design trends, and their impact on next generation routers. We also describe router architectures and the challenges involved in designing high-performance large-scale routers.

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

1.1 ARCHITECTURE OF THE INTERNET: PRESENT AND FUTURE

1.1.1 The Present

Today's Internet is an amalgamation of thousands of commercial and service provider networks. It is not feasible for a single service provider to connect two distant nodes on the Internet. Therefore, service providers often rely on each other to connect the dots. Depending on the size of network they operate, Internet Service Providers (ISPs) can be broken down into three major categories. Tier-1 ISPs are about a dozen major telecommunication companies, such as UUNet, Sprint, Qwest, XO Network, and AT&T, whose high-speed global networks form the Internet backbone. Tier-1 ISPs do not buy network capacity from other providers; instead, they sell or lease access to their backbone resource to smaller Tier-2 ISPs, such as America Online and Broadwing. Tier-3 ISPs are typically regional service providers such as Verizon and RCN through whom most enterprises connect to the Internet. Figure 1.1 illustrates the architecture of a typical Tier-1 ISP network.

Each Tier-1 ISP operates multiple IP/MPLS (multi-protocol label switching), and sometimes ATM (asynchronous transfer mode), backbones with speeds varying anywhere from T3 to OC-192 (optical carrier level 192, ~10 Gbps). These backbones are interconnected through peering agreements between ISPs to form the Internet backbone. The backbone is designed to transfer large volumes of traffic as quickly as possible between networks. Enterprise networks are often linked to the rest of the Internet via a variety of links, anywhere from a T1 to multiple OC-3 lines, using a variety of Layer 2 protocols, such as Gigabit Ethernet, frame relay, and so on. These enterprise networks are then overhauled into service provider networks through edge routers. An edge router can aggregate links from multiple enterprises. Edge routers are interconnected in a pool, usually at a Point of Presence (POP)

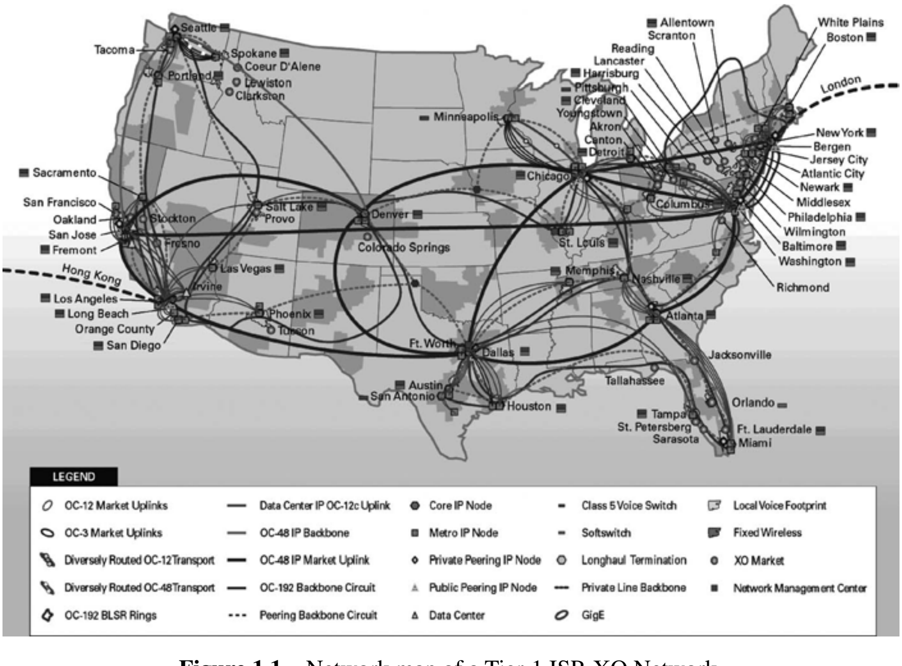

Figure 1.1 Network map of a Tier-1 ISP, XO Network.

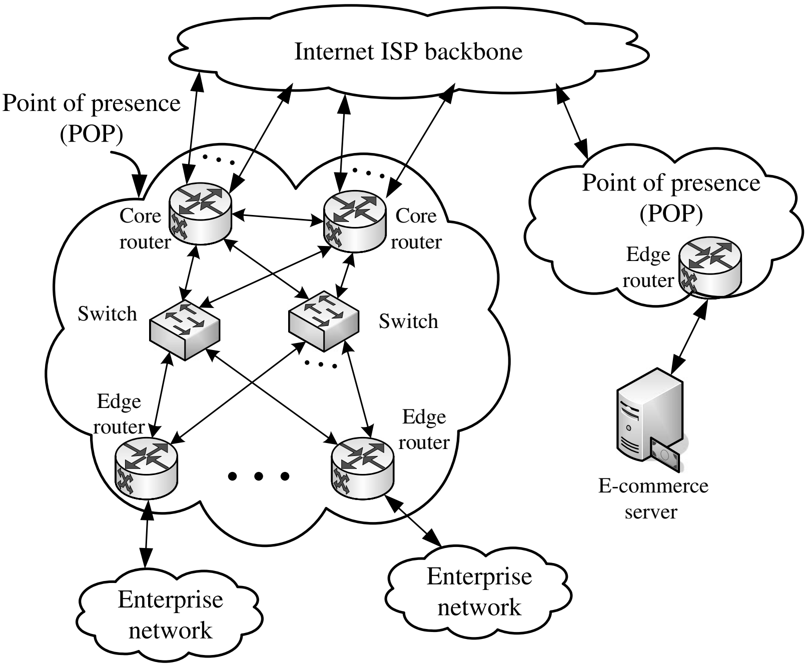

Figure 1.2 Point of presence (POP).

of a service provider, as shown in Figure 1.2. Each POP may link to other POPs of the same ISP through optical transmission/switching equipment, may link to POPs of other ISPs to form a peering, or link to one or more backbone routers. Typically, a POP may have a few backbone routers in a densely connected mesh. In most POPs, each edge router connects to at least two backbone routers for redundancy. These backbone routers may also connect to backbone routers at other POPs according to ISP peering agreements. Peering occurs when ISPs exchange traffic bound for each other's network over a direct link without any fees. Therefore, peering works best when peers exchange roughly the same amount of traffic. Since smaller ISPs do not have high quantities of traffic, they often have to buy *transit* from a Tier-1 provider to connect to the Internet. A recent study of the topologies of 10 service providers across the world shows that POPs share this generic structure [3].

Unlike POPs, the design of backbone varies from service provider to service provider. For example, Figure 1.3 illustrates backbone design paradigms of three major service providers

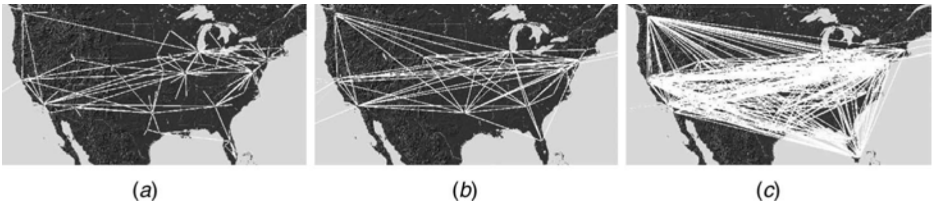

Figure 1.3 Three distinct backbone design paradigms of Tier-1 ISPs. (*a*) AT&T; (*b*) Sprint; (*c*) Level 3 national network infrastructure [3].

in the US. AT&T's backbone design includes large POPs at major cities, which in turn fan out into smaller per-city POPs. In contrast, Sprint's backbone has only 20 well connected POPs in major cities and suburban links are back-hauled into the POPs via smaller ISPs. Most major service providers still have the AT&T backbone model and are in various stages of moving to Sprint's design. Sprint's backbone design provides a good solution to service providers grappling with a need to reduce capital expenditure and operational costs associated with maintaining and upgrading network infrastructure. Interestingly, Level 3 presents another design paradigm in which the backbone is highly connected via circuit technology such as, MPLS, ATM or frame relays. As will be seen later, this is the next generation of network design where the line between backbone and network edge begins to blur.

Now, let us see how network design impacts on the next generation routers. Router design is often guided by the economic requirements of service providers. Service providers would like to reduce the infrastructure and maintenance costs while, at the same time, increasing available bandwidth and reliability. To this end, network backbone has a set of well-defined, narrow requirements. Routers in the backbone should simply move traffic as fast as possible. Network edge, however, has broad and evolving requirements due simply to the diversity of services and Layer 2 protocols supported at the edge. Today most POPs have multiple edge routers optimized for point solutions. In addition to increasing infrastructure and maintenance costs, this design also increases the complexity of POPs resulting in an unreliable network infrastructure. Therefore, newer edge routers have been designed to support diversity and are easily adaptable to the evolving requirements of service providers. This design trend is shown in Table 1.1, which lists some properties of enterprise, edge, and core routers currently on the market. As we will see in the following sections, future network designs call for the removal of edge routers altogether and their replacement with fewer core routers to increase reliability, throughput, and to reduce costs. This means next generation routers would have to amalgamate the diverse service requirements of edge routers and the strict performance requirements of core routers, seamlessly into one body. Therefore, the real question is not whether we should build highly-flexible, scalable, high-performance routers, but how?

1.1.2 The Future

As prices of optical transport and optical switching sharply decrease, some network designers believe that the future network will consist of many mid-size IP routers or MPLS

TABLE 1.1 Popular Enterprise, Edge, and Core Routers in the Market

Model	Capacity[a]	Memory	Power	Features
Cisco 7200	–	256 MB	370 W	QoS, MPLS, Aggregation
Cisco 7600	720 Gbps	1 GB	–	QoS, MPLS, Shaping
Cisco 10000	51.2 Gbps	–	1200 W	QoS, MPLS
Cisco 12000	1.28 Tbps	4 GB	4706 W	MPLS, Peering
Juniper M-320	320 Gbps	2 GB	3150 W	MPLS, QoS, VPN
Cisco CRS	92 Tbps	4 GB	16,560 W	MPLS, Qos, Peering
Juniper TX/T-640	2.5 Tbps/640 Gbps	2 GB	4550 W/6500 W	MPLS, QoS, Peering

[a] Note that the listed capacity is the combination of ingress and egress capacities.

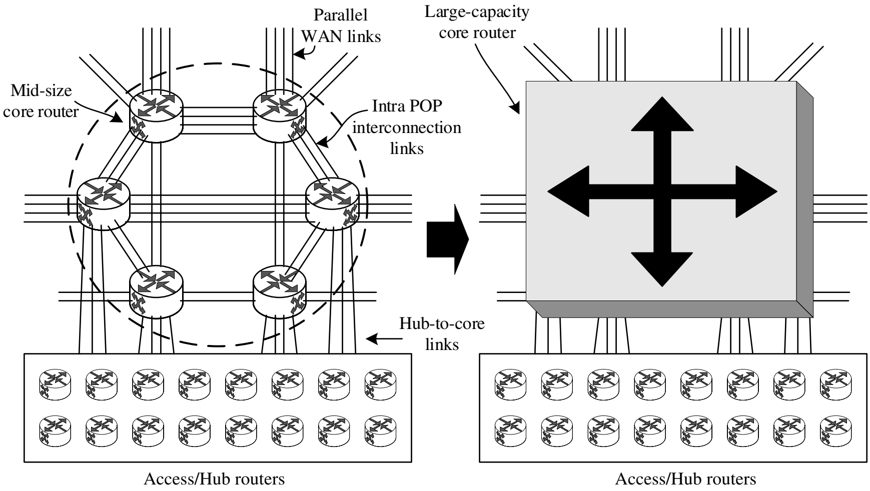

Figure 1.4 Replacing a cluster of mid-size routers with a large-capacity scalable router.

switches at the network edge that are connected to optical crossconnects (OXCs), which are then interconnected by DWDM transmission equipment. The problem for this approach is that connections to the OXC are usually high bit rates, for example, 10 Gbps for now and 40 Gbps in the near future. When the edge routers want to communicate with all other routers, they either need to have direct connections to those routers or connect through multiple logical hops (i.e., routed by other routers). The former case results in low link utilization while the latter results in higher latency. Therefore, some network designers believe it is better to build very large IP routers or MPLS switches at POPs. They aggregate traffic from edge routers onto high-speed links that are then directly connected to other large routers at different POPs through DWDM transmission equipment. This approach achieves higher link utilization and fewer hops (thus lower latency). As a result, the need for an OXC is mainly for provisioning and restoring purposes but not for dynamic switching to achieve higher link utilization.

Current router technologies available in the market cannot provide large switching capacities to satisfy current and future bandwidth demands. As a result, a number of mid-size core routers are interconnected with numerous links and use many expensive line cards that are used to carry intra-cluster traffic rather than revenue-generating users' or wide-area-network (WAN) traffic. Figure 1.4 shows how a router cluster is replaced by a large-capacity scalable router, saving the cost of numerous line cards and links, and real estate. It provides a cost-effective solution that can satisfy Internet traffic growth without having to replace routers every two to three years. Furthermore, there are fewer individual routers that need to be configured and managed, resulting in a more efficient and reliable system.

1.2 ROUTER ARCHITECTURES

IP routers' functions can be classified into two categories: datapath functions and control plane functions [4].

The datapath functions such as forwarding decision, forwarding through the backplane, and output link scheduling are performed on every datagram that passes through the router. When a packet arrives at the forwarding engine, its destination IP address is first masked by the subnet mask (logical AND operation) and the resulting address is used to lookup the forwarding table. A so-called longest prefix matching method is used to find the output port. In some applications, packets are classified based on 104 bits that include the IP source/destination addresses, transport layer port numbers (source and destination), and type of protocol, which is generally called 5-tuple. Based on the result of classification, packets may be either discarded (firewall application) or handled at different priority levels. Then, time-to-live (TTL) value is decremented and a new header checksum is recalculated.

The control plane 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, for example, RIP (Routing Information Protocol), OSPF (Open Shortest Path Forwarding), or BGP (Border Gateway Protocol). It can also create a forwarding table for the forwarding engine. Since the control functions are not performed on each arriving individual packet, they do not have a strict speed constraint and are implemented in software in general.

Router architectures generally fall into two categories: centralized (Fig. 1.5*a*) and distributed (Fig. 1.5*b*).

Figure 1.5*a* shows a number of network interfaces, forwarding engines, a route controller (RC), and a management controller (MC) interconnected by a switch fabric. Input interfaces send packet headers to the forwarding engines through the switch fabric. The forwarding engines, in turn, determine which output interface the packet should be sent to. This information is sent back to the corresponding input interface, which forwards the packet to the right output interface. The only task of a forwarding engine is to process packet headers and is shared by all the interfaces. All other tasks such as participating in routing protocols, reserving resource, handling packets that need extra attention, and other administrative and maintenance tasks, are handled by the RC and the MC. The BBN multi-gigabit router [5] is an example of this design.

The difference between Figure 1.5*a* and 1.5*b* is that the functions of the forwarding engines are integrated into the interface cards themselves. Most high-performance routers use this architecture. The RC maintains a routing table and updates it based on routing protocols used. The routing table is used to generate a forwarding table that is then downloaded

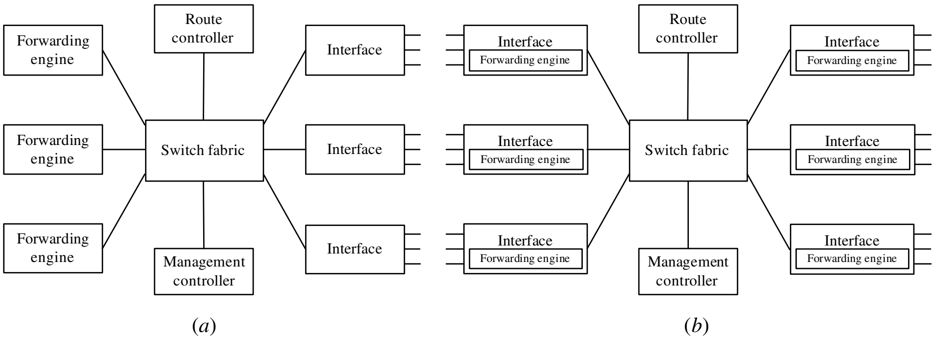

Figure 1.5 (*a*) Centralized versus (*b*) distributed models for a router.

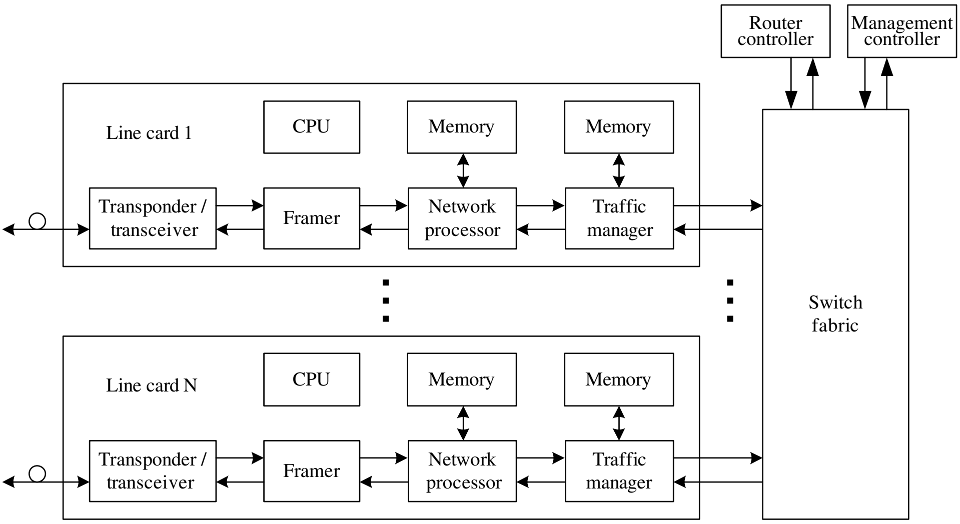

Figure 1.6 Typical router architecture.

from the RC to the forwarding engines in the interface cards. It is not necessary to download a new forwarding table for every route update. Route updates can be frequent, but routing protocols need time, in the order of minutes, to converge. The RC needs a dynamic routing table designed for fast updates and fast generation of forwarding tables. Forwarding tables, on the other hand, can be optimized for lookup speed and need not be dynamic.

Figure 1.6 shows a typical router architecture, where multiple line cards, an RC, and an MC are interconnected through a switch fabric. The communication between the RC/MC and the line cards can be either through the switch fabric or through a separate interconnection network, such as a Ethernet switch. The line cards are the entry and exit points of data to and from a router. They provide the interface from physical and higher layers to the switch fabric. The tasks provided by line cards are becoming more complex as new applications develop and protocols evolve. Each line card supports at least one full-duplex fiber connection on the network side, and at least one ingress and one egress connection to the switch fabric backplane. Generally speaking, for high-bandwidth applications, such as OC-48 and above, the network connections support channelization for aggregation of lower-speed lines into a large pipe, and the switch fabric connections provide flow-control mechanisms for several thousand input and output queues to regulate the ingress and egress traffic to and from the switch fabric.

A line card usually includes components such as a transponder, framer, network processor (NP), traffic manager (TM), and central processing unit (CPU).

Transponder/Transceiver. This component performs optical-to-electrical and electrical-to-optical signal conversions, and serial-to-parallel and parallel-to-serial conversions [6, 7].

Framer. A framer performs synchronization, frame overhead processing, and cell or packet delineation. On the transmit side, a SONET (synchronous optical network)/SDH (synchronous digital hierarchy) framer generates section, line, and path overhead. It performs framing pattern insertion (A1, A2) and scrambling. It generates section, line, and path bit interleaved parity (B1/B2/B3) for far-end performance monitoring. On the receive side, it processes section, line, and path overhead. It performs frame delineation, descrambling, alarm detection, pointer interpretation, bit interleaved parity monitoring (B1/B2/B3), and error count accumulation for performance monitoring [8]. An alternative for the framer is Ethernet framer.

Network Processor. The NP mainly performs table lookup, packet classification, and packet modification. Various algorithms to implement the first two functions are presented in Chapters 2 and 3, respectively. The NP can perform those two functions at the line rate using external memory, such as static random access memory (SRAM) or dynamic random access memory (DRAM), but it may also require external content addressable memory (CAM) or specialized co-processors to perform deep packet classification at higher levels. In Chapter 16, we present some commercially available NP and ternary content addressable memory (TCAM) chips.

Traffic Manager. To meet the requirements of each connection and service class, the TM performs various control functions to cell/packet streams, including traffic access control, buffer management, and cell/packet scheduling. Traffic access control consists of a collection of specification techniques and mechanisms that (1) specify the expected traffic characteristics and service requirements (e.g., peak rate, required delay bound, loss tolerance) of a data stream; (2) shape (i.e., delay) data streams (e.g., reducing their rates and/or burstiness); and (3) police data streams and take corrective actions (e.g., discard, delay, or mark packets) when traffic deviates from its specification. The usage parameter control (UPC) in ATM and differentiated service (DiffServ) in IP performs similar access control functions at the network edge. Buffer management performs cell/packet discarding, according to loss requirements and priority levels, when the buffer exceeds a certain threshold. Proposed schemes include early packet discard (EPD) [9], random early packet discard (REPD) [10], weighted REPD [11], and partial packet discard (PPD) [12]. Packet scheduling ensures that packets are transmitted to meet each connection's allocated bandwidth/delay requirements. Proposed schemes include deficit round-robin, weighted fair queuing (WFQ) and its variants, such as shaped virtual clock [13] and worst-case fairness WFQ (WF^{2}Q+) [14]. The last two algorithms achieve the worst-case fairness properties. Details are discussed in Chapter 4. Many quality of service (QoS) control techniques, algorithms, and implementation architectures can be found in Ref. [15]. The TM may also manage many queues to resolve contention among the inputs of a switch fabric, for example, hundreds or thousands of virtual output queues (VOQs). Some of the representative TM chips on the market are introduced in Chapter 16, whose purpose it is to match the theories in Chapter 4 with practice.

Central Processing Unit. The CPU performs control plane functions including connection set-up/tear-down, table updates, register/buffer management, and exception handling. The CPU is usually not in-line with the fast-path on which maximum-bandwidth network traffic moves between the interfaces and the switch fabric.

The architecture in Figure 1.6 can be realized in a multi-rack (also known as multi-chassis or multi-shelf) system as shown in Figure 1.7. In this example, a half rack, equipped with a switch fabric, a duplicated RC, a duplicated MC, a duplicated system clock (CLK), and a duplicated fabric shelf controller (FSC), is connected to all other line card (LC) shelves, each of which has a duplicated line card shelf controller (LSC). Both the FSC and the

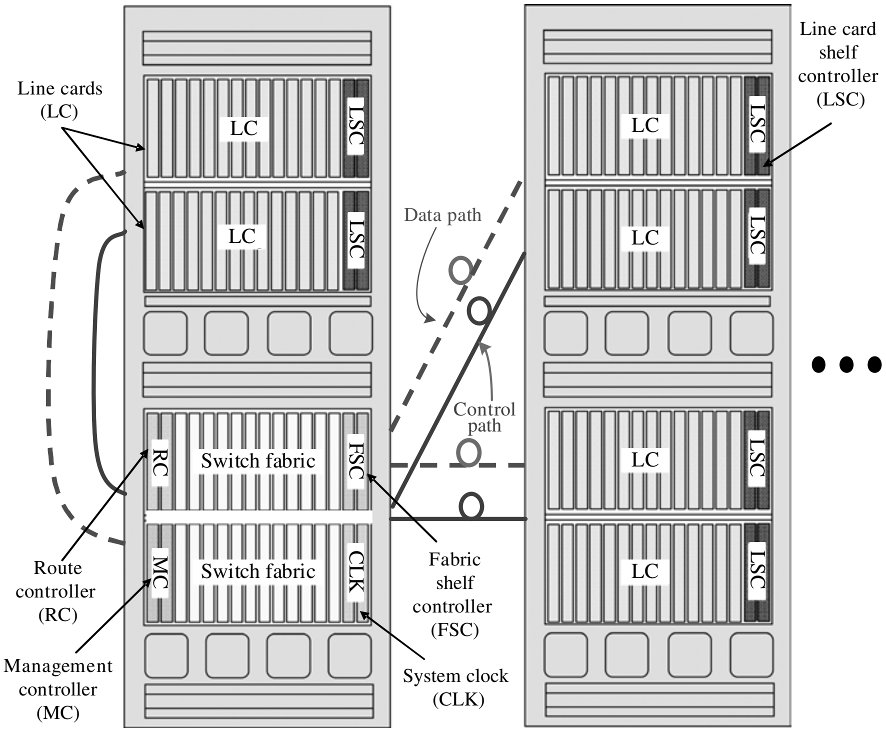

Figure 1.7 Multi-rack router system.

LSC provide local operation and maintenance for the switch fabric and line card shelves, respectively. They also provide the communication channels between the switch/line cards with the RC and the MC. The duplicated cards are for reliability concerns. The figure also shows how the system can grow by adding more LC shelves. Interconnections between the racks are sets of cables or fibers, carrying information for the data and the control planes. The cabling usually is a combination of unshielded twisted path (UTP) Category 5 Ethernet cables for control path, and fiber-optic arrays for data path.

1.3 COMMERCIAL CORE ROUTER EXAMPLES

We now briefly discuss the two most popular core routers on the market: Juniper Network's T640 TX-Matrix [16] and Cisco System's Carrier Routing System (CRS-1) [17].

1.3.1 T640 TX-Matrix

A T640 TX-Matrix is composed of up to four routing nodes and a TX Routing Matrix interconnecting the nodes. A TX Routing Matrix connects up to four T640 routing nodes via a three-stage Clos network switch fabric to form a unified router with the capacity of 2.56 Terabits. The blueprint of a TX Routing Matrix is shown in Figure 1.8. The unified router is controlled by the Routing Engine of the matrix which is responsible for running routing protocols and for maintaining overall system state. Routing engines in each routing

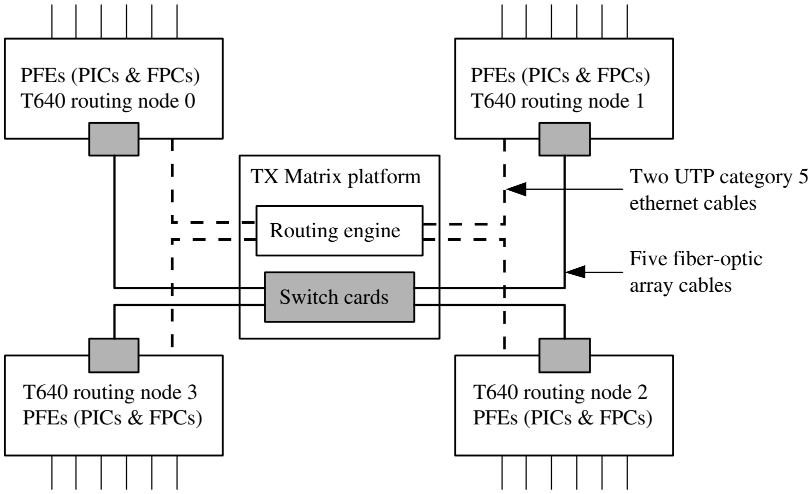

Figure 1.8 TX Routing Matrix with four T640 routing nodes.

node manage their individual components in coordination with the routing engine of the matrix. Data and control plane of each routing node is interconnected via an array of optical and Ethernet cables. Data planes are interconnected using VCSEL (vertical cavity surface emitting laser) optical lines whereas control planes are interconnected using UTP Category 5 Ethernet cables.

As shown in Figure 1.9, each routing node has two fundamental architectural components, namely the control plane and the data plane. The T640 routing node's control plane is implemented by the JUNOS software that runs on the node's routing engine. JUNOS is a micro-kernel-based modular software that assures reliability, fault isolation, and high availability. It implements the routing protocols, generates routing tables and forwarding tables, and supports the user interface to the router. Data plane, on the other hand, is responsible for processing packets in hardware before forwarding them across the switch fabric from the ingress interface to the appropriate egress interface. The T640 routing node's data plane is implemented in custom ASICs in a distributed architecture.

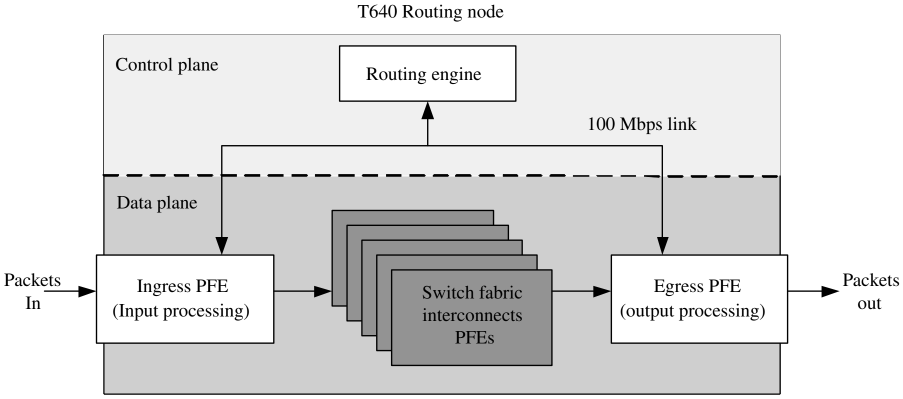
T640 Routing node
Control plane
Routing engine
100 Mbps link
Data plane
Packets In
Ingress PFE (Input processing)
Switch fabric interconnects PFEs
Egress PFE (output processing)
Packets out

Figure 1.9 T640 routing node architecture.

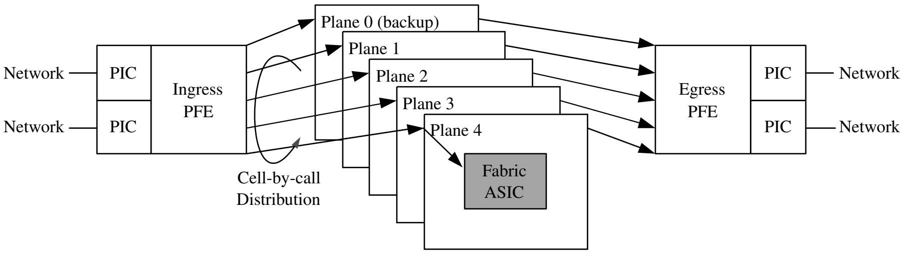

Figure 1.10 T640 switch fabric planes.

The T640 routing node has three major elements: Packet forwarding engines (PFEs), the switch fabric, and one or two routing engines. The PFE performs Layer 2 and Layer 3 packet processing and forwarding table lookups. A PFE is made of many ASIC components. For example, there are Media-Specific ASICs to handle Layer 2 functions that are associated with the specific physical interface cards (PICs), such as SONET, ATM, or Ethernet. L2/L3 Packet Processing, and ASICs strip off Layer 2 headers and segment packets into cells for internal processing, and reassemble cells into Layer 3 packets prior to transmission on the egress interface. In addition, there are ASICs for managing queuing functions (Queuing and Memory Interface ASIC), for forwarding cells across the switch fabric (Switch Interface ASICs), and for forwarding lookups (T-Series Internet Processor ASIC).

The switch fabric in a standalone T640 routing node provides data plane connectivity among all of the PFEs in the chassis. In a TX-Routing Matrix, switch fabric provides data plane connectivity among all of the PFEs in the matrix. The T640 routing node uses a Clos network and the TX-Routing Matrix uses a multistage Clos network. This switch fabric provides nonblocking connectivity, fair bandwidth allocation, and distributed control. In order to achieve high-availability each node has up to five switch fabric planes (see Fig. 1.10). At a given time, four of them are used in a round-robin fashion to distribute packets from the ingress interface to the egress interface. The fifth one is used as a hot-backup in case of failures. Access to switch fabric bandwidth is controlled by the following three-step request-grant mechanism. The request for each cell of a packet is transmitted in a round-robin order from the source PFE to the destination PFE. Destination PFE transmits a grant to the source using the same switch plane from which the corresponding request was received. Source PFE then transmits the cell to the destination PFE on the same switch plane.

1.3.2 Carrier Routing System (CRS-1)

Cisco System's Carrier Routing System is shown in Figure 1.11. CRS-1 also follows the multi-chassis design with line card shelves and fabric shelves. The design allows the system to combine as many as 72 line card shelves interconnected using eight fabric shelves to operate as a single router or as multiple logical routers. It can be configured to deliver anywhere between 1.2 to 92 terabits per second capacity and the router as a whole can accommodate 1152 40-Gbps interfaces. Router Engine is implemented using at least two route processors in a line card shelf. Each route processor is a Dual PowerPC CPU complex configured for symmetric multiprocessing with 4 GB of DRAM for system processes and routing tables and 2 GB of Flash memory for storing software images and system configuration. In addition, the system is equipped to include non-volatile random access

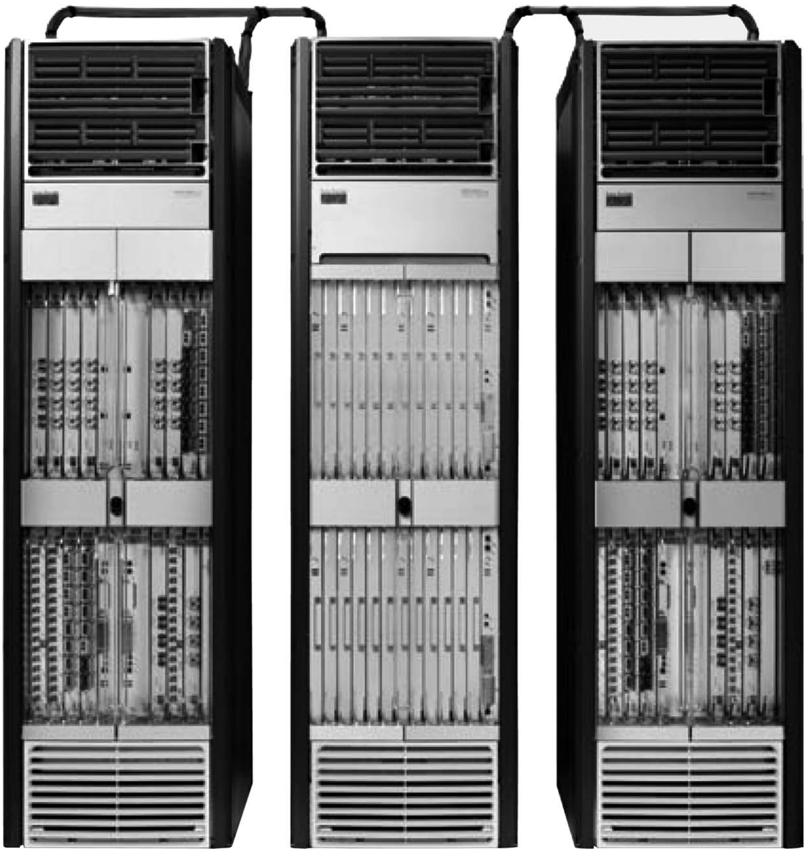

Figure 1.11 Cisco CRS-1 carrier routing system.

memory (NVRAM) for configurations and logs and a 40 GB on-board hard drive for data collection. Data plane forwarding functions are implemented through Cisco's Silicon Packet Processor (SPP), an array of 188 programmable reduced instruction set computer (RISC) processors.

Cisco CRS-1 uses a three-stage, dynamically self-routed Benes topology based switching fabric. A high-level diagram of the switch fabric is shown in Figure 1.12. The first-stage (S1) of the switch is connected to ingress line cards. Stage-2 (S2) fabric cards receive cells from Stage-1 fabric cards and deliver them to Stage-3 fabric cards that are associated with the appropriate egress line cards. Stage-2 fabric cards support speedup and multicast replication. The system has eight such switch fabrics operating in parallel through which cells are transferred evenly. This fabric configuration provides highly scalable, available, and survivable interconnections between the ingress and egress slots. The whole system is driven by a Cisco Internet Operating System (IOS) XR. The Cisco IOS XR is built on a micro-kernel-based memory-protected architecture, to be modular. This modularity provides for better scalability, reliability, and fault isolation. Furthermore, the system implements check pointing and stateful hot-standby to ensure that critical processes can be restarted with minimal effect on system operations or routing topology.

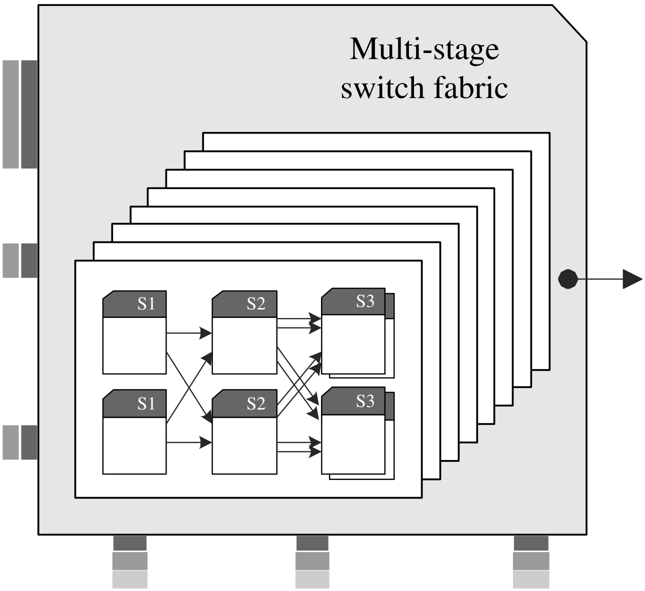

Figure 1.12 High-level diagram of Cisco CRS-1 multi-stage switch fabric.

1.4 DESIGN OF CORE ROUTERS

Core routers are designed to move traffic as quickly as possible. With the introduction of diverse services at the edges and rapidly increasing bandwidth requirements, core routers now have to be designed to be more flexible and scalable than in the past. To this end, design goals of core routers generally fall into the following categories:

Packet Forwarding Performance. Core routers need to provide packet forwarding performance in the range of hundreds of millions of packets per second. This is required to support existing services at the edges, to grow these services in future, and to facilitate the delivery of new revenue-generating services.

Scalability. As the traffic rate at the edges grows rapidly, service providers are forced to upgrade their equipment every three to five years. Latest core routers are designed to scale well such that subsequent upgrades are cheaper to the providers. To this end, the latest routers are designed as a routing matrix to add future bandwidth while keeping the current infrastructure in place. In addition, uniform software images and user interfaces across upgrades ensure the users do not need to be retrained to operate the new router.

Bandwidth Density. Another issue with core routers is the amount of real estate and power required to operate them. Latest core routers increase bandwidth density by providing higher bandwidths in small form-factors. For example, core routers that provide 32 × OC-192 or 128 × OC-48 interfaces in a half-rack space are currently available on the market. Such routers consume less power and require less real estate.

Service Delivery Features. In order to provide end-to-end service guarantees, core routers are also required to provide various services such as aggregate DiffServ classes, packet filtering, policing, rate-limiting, and traffic monitoring at high speeds.

These services must be provided by core routers without impacting packet forwarding performance.

Availability. As core routers form a critical part of the network, any failure of a core router can impact networks dramatically. Therefore, core routers require higher availability during high-traffic conditions and during maintenance. Availability on most core routers is achieved via redundant, hot-swappable hardware components, and modular software design. The latest core routers allow for hardware to be swapped out and permit software upgrades while the system is on-line.

Security. As the backbone of network infrastructure, core routers are required to provide some security related functions as well. Besides a secure design and implementation of their own components against denial of service attacks and other vulnerabilities, the routers also provide rate-limiting, filtering, tracing, and logging to support security services at the edges of networks.

It is very challenging to design a cost-effective large IP router with a capacity of a few hundred terabits/s to a few petabit/s. Obviously, the complexity and cost of building a large-capacity router is much higher than building an OXC. This is because, for packet switching, there is a requirement to process packets (such as classification, table lookup, and packet header modification), store them, schedule them, and perform buffer management. As the line rate increases, the processing and scheduling time associated with each packet is proportionally reduced. Also, as the router capacity increases, the time interval for resolving output contention becomes more constrained. Memory and interconnection technologies are the most demanding when designing a large-capacity packet switch. The former very often becomes a bottleneck for a large-capacity packet switch while the latter significantly affects a system's power consumption and cost. As a result, designing a cost-effective, large capacity switch architecture still remains a challenge. Several design issues are discussed below.

Memory Speed. As optical and electronic devices operate at 10 Gbps (OC-192) at present, the technology and the demand for optical channels operating at 40 Gbps (OC-768) is a emerging. The port speed to a switch fabric is usually twice that of the line speed. This is to overcome some performance degradation that otherwise arises due to output port contention and the overhead used to carry routing, flow control, and QoS information in the packet/cell header. As a result, the aggregated I/O bandwidth of the memory at the switch port can be 120 Gbps. Considering 40-byte packets, the cycle time of the buffer memory at each port is required to be less than 2.66 ns. This is still very challenging with current memory technology, especially when the required memory size is very large and cannot be integrated into the ASIC (application specific integrated circuit), such as for the traffic manager or other switch interface chips. In addition, the pin count for the buffer memory can be several hundreds, limiting the number of external memories that can be attached to the ASIC.

Packet Arbitration. An arbitrator is used to resolve output port contention among the input ports. Considering a 40-Gbps switch port with 40-byte packets and a speedup of two, the arbitrator has only about 4 ns to resolve the contention. As the number of input ports increases, the time to resolve the contention reduces. It can be implemented in a centralized way, where the interconnection between the arbitrator and all input line (or port) cards can be prohibitively complex and expensive. On the other hand, it can be implemented in a distributed way, where the line cards and switch cards are involved in the arbitration. The distributed implementation may degrade throughput and delay performance due to lack of the availability of the state information of all inputs and outputs. As a result, a higher speedup is required in the switch fabric to improve performance.

QoS Control. Similar to the above packet arbitration problem, as the line (port) speed increases, the execution of policing/shaping at the input ports and packet scheduling and buffer management (discarding packet policies) at the output port (to meet the QoS requirement of each flow or each class) can be very difficult and challenging. The buffer size at each line card is usually required to hold up to 100 ms worth of packets. For a 40-Gbps line, the buffer can be as large as 500 Mbytes, which can store hundreds of thousands of packets. Choosing a packet to depart or to discard within 4 to 8 ns is not trivial. In addition, the number of states that need to be maintained to do per-flow control can be prohibitively expensive. An alternative is to do class-based scheduling and buffer management, which is more sensible at the core network, because the number of flows and the link speed is too high. Several shaping and scheduling schemes require time stamping arriving packets and scheduling their departure based on the time stamp values. Choosing a packet with the smallest time stamp in 4 to 8 ns can cause a bottleneck.

Optical Interconnection. A large-capacity router usually needs multiple racks to house all the line cards, port cards (optional), switch fabric cards, and controller cards, such as route controller, management controller, and clock distribution cards. Each rack may accommodate 0.5 to 1 terabit/s capacity depending on the density of the line and switch fabric cards and may need to communicate with another rack (e.g., the switch fabric rack) with a bandwidth of 0.5 to 1.0 terabit/s in each direction. With current VCSEL technology, an optical transceiver can transmit up to 300 meters with 12 SERDES (serializer/deserializer) channels, each running at 2.5 or 3.125 Gbps [18]. They have been widely used for backplane interconnections. However, the size and power consumption of these optical devices could limit the number of interconnections on each circuit board, resulting in more circuit boards, and thus higher implementation costs. Furthermore, a large number of optical fibers are required to interconnect multiple racks. This increases installation costs and makes fiber reconfiguration and maintenance difficult. The layout of fiber needs to be carefully designed to reduce potential interruption caused by human error. Installing new fibers to scale the router's capacity can be mistake-prone and disrupting to the existing services.

Power Consumption. As SERDES technology allows more than a hundred bi-directional channels, each operating at 2.5 or 3.125 Gbps, on a CMOS (complementary metal-oxide-semiconductor) chip [19, 20], its power dissipation can be as high as 20 W. With VCSEL technology, each bi-directional connection can consume 250 mW. If we assume that 1 terabit/s bandwidth is required for interconnection to other racks, it would need 400 optical bi-directional channels (each 2.5 Gbps), resulting in a total of 1000 W per rack for optical interconnections. Each rack may dissipate up to several thousands watts due to the heat dissipation limitation, which in turn limits the number of components that can be put on each card and limits the number of cards on each rack. The large power dissipation also increases the cost of air-conditioning the room. The power consumption cannot be overlooked from the global viewpoint of the Internet [21].

Flexibility. As we move the core routers closer to the edge of networks, we now have to support diverse protocols and services available at the edge. Therefore, router design must be modular and should evolve with future requirements. This means we cannot rely too heavily on fast ASIC operations; instead a balance needs to be struck between performance and flexibility by ways of programmable ASICs.

1.5 IP NETWORK MANAGEMENT

Once many switches and routers are interconnected on the Internet, how are they managed by the network operators? In this section, we briefly introduce the functionalities, architecture, and major components of the management systems for IP networks.

1.5.1 Network Management System Functionalities

In terms of the network management model defined by the International Standard Organization (ISO), a network management system (NMS) has five management functionalities [22–24]: performance management (PM), fault management (FM), configuration management (CM), accounting management (AM), and security management (SM).

PM. The task of PM is to monitor, measure, report, and control the performance of the network, which can be done by monitoring, measuring, reporting, and controlling the performance of individual network elements (NEs) at regular intervals; or by analyzing logged performance data on each NE. The common performance metrics are network throughput, link utilization, and packet counts input and output from an NE.

FM. The goal of FM is to collect, detect, and respond to fault conditions in the network, which are reported as trap events or alarm messages. These messages may be generated by a managed object or its agent built into a network device, such as Simple Network Management Protocol (SNMP) traps [25] or Common Management Information Protocol (CMIP) event notifications [26, 27], or by a network management system (NMS), using synthetic traps or probing events generated by, for instance, Hewlett-Packard's OpenView (HPOV) stations. Fault management systems handle network failures, including hardware failures, such as link down and software failures, and protocol errors, by generating, collecting, processing, identifying, and reporting trap and alarm messages.

CM. The task of CM includes configuring the switch and I/O modules in a router, the data and management ports in a module, and the protocols for a specific device. CM deals with the configuration of the NEs in a network to form a network and to carry customers' data traffic.

AM. The task of AM is to control and allocate user access to network resources, and to log usage information for accounting purposes. Based on the price model, logged information, such as call detailed records (CDR), is used to provide billing to customers. The price model can be usage-based or flat rate.

SM. SM deals with protection of network resources and customers' data traffic, including authorization and authentication of network resources and customers, data integrity, and confidentiality. Basic access control to network resources by using login and password, generation of alarms for security violation and authorization failure, definition and enforcement of security policy, and other application layer security measures such as firewalls, all fall under the tasks of security management.

1.5.2 NMS Architecture

Within a network with heterogeneous NEs, the network management tools can be divided into three levels: element management system (EMS), from network equipment vendors that specialize in the management of the vendor's equipment; NMS, aimed at managing networks with heterogeneous equipment; and operational support systems (OSS), operating support and managing systems developed for network operator's specific operations, administration, and maintenance (OAM) needs. A high-level view of the architecture of a typical NMS is shown in Figure 1.13. In this architecture, the management data are collected and processed in three levels.

EMS Level. Each NE has its own EMS, such as EMS1, EMS2, and EMS3, shown in Figure 1.13. These EMS collect management data from each NE, process the data, and forward the results to the NMS that manages the overall network. In this way, the EMS and NMS form a distributed system architecture.

NMS Level. Functionally, an NMS is the same as an EMS, except an NMS has to deal with many heterogeneous NEs. The NMS station gathers results from the EMS

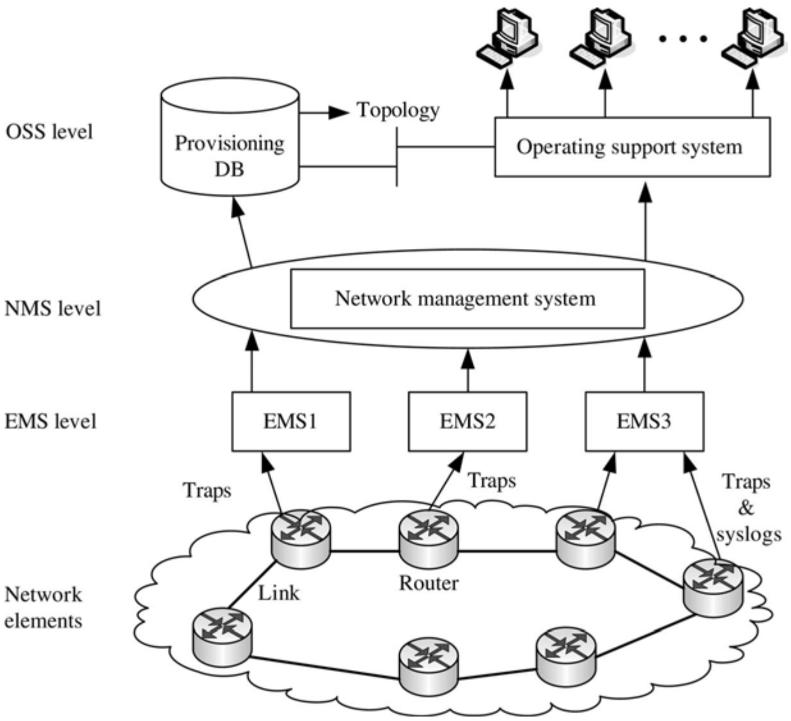

Figure 1.13 An NMS architecture.

stations, displays information, and takes control actions. For example, an NMS aggregates the events from all the related NEs in handling a specific fault condition to identify the root cause and to reduce the number of events that are sent to the OSS for further processing. Note that the NMS is independent of specific NEs.

OSS Level. By combing the network topology information, the OSS further collects and processes the data for specific operational needs. Therefore, the OSS can have subsystems for PM, FM, AM, and SM.

A key feature of this architecture is that each of the three levels performs all of the network management functions by generating, collecting, processing, and logging the events to solve the scalability issues in large-scale networks.

There are many NMS tools that are commercially available [28, 29]. For example, Cisco's IOS for the management of LANs (local area networks) and WANs (wide area networks) built on Cisco switches and routers; and Nortel's Optivity NMS for the management of Nortel's ATM switches and routers. To manage networks with heterogeneous NEs, the available tools are HPOV, Node Manager, Aprisma's SPECTRUM, and Sun's Solstice NMS. These tools support SNMP and can be accessed through a graphical user interface (GUI) and command line interface (CLI). Some of them also provide automated assistance for CM and FM tasks.

1.5.3 Element Management System

As a generic solution for configuring network devices, monitoring status, and checking devices for errors, the Internet-standard framework for network management is used for the management tasks of an NE, as for an IP network. Therefore, functionally, an EMS and NMS have the same architectures. The same five functions for network management are also used for element functions.

The architecture of a general EMS is shown in Figure 1.14. On the device side, the device must be manageable, that is, it must have a management agent such as the SNMP agent (or server), corresponding data structures, and a storage area for the data. On the EMS station side, the station must have a management client such as the SNMP manager (or client). In between the management station and the managed device, we also need a protocol for the communications of the two parties, for example, SNMP.

The core function to manage a device is implemented by using an SNMP manager. Whenever there is a command issued by a user through the user interface, the command is received by the SNMP manager after parsing. If it is a configure command, the SNMP manager issues an SNMP request to the SNMP agent inside the device. From the device, the SNMP agent then goes to the management information bases (MIBs) to change the value of a specified MIB object. This is shown as 'Config' in Figure 1.14. Config can be done by a simple command such as 'set'.

Similarly, if the command issued by the user is to get the current status of the device, the SNMP manager issues an SNMP request to the SNMP agent inside the device. From the device, the SNMP agent then goes to the MIBs to get the value of a specified MIB object by a 'get' command, which is shown as 'View' in Figure 1.14. Then, the SNMP agent forwards the obtained MIB values to the SNMP manager as response back. The response is finally sent to the user for display on the GUI or CLI console.

In some cases, the device may send out messages to its SNMP agent autonomously. One example is the trap or alarm, where the initiator of the event is not the user interface but

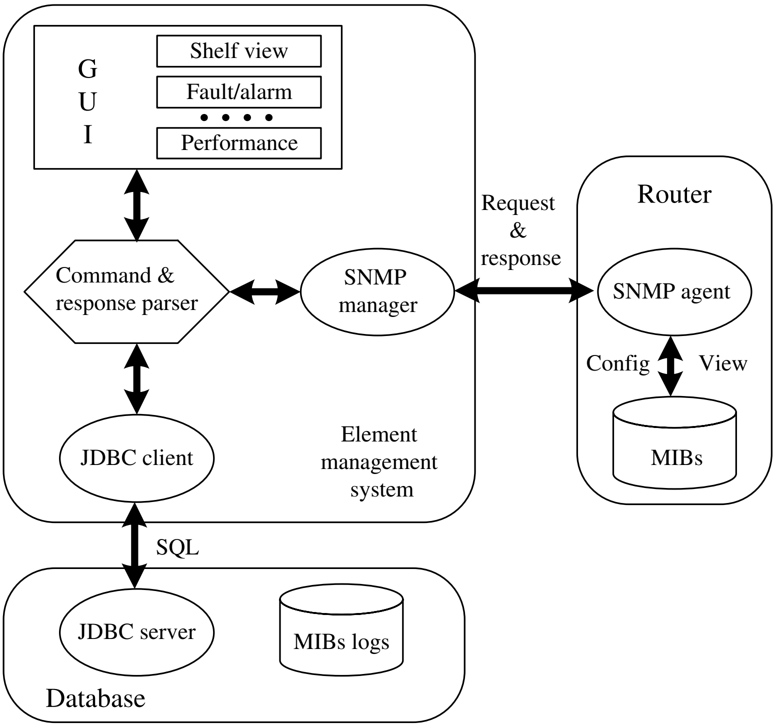

Figure 1.14 EMS archtiecture.

the device. Here, the most important communications are regulated by the SNMP protocol, including the operations and protocol data unit (PDU) format.

Note that all the configuration data and performance statistics are usually saved in a separate database. For example, for disaster recovery purposes, the changes in the configuration of a device will also be saved in the database. The database saves both MIB information and log messages. The communications between the database and the management client are implemented by using a database client inside the management client and database server inside the database. As shown in Figure 1.14, a popular choice is a JDBC (Java Database Connectivity) client and a JDBC server in the two sides. The commands and responses between the EMS and the device are parsed and converted into structured query language (SQL) commands to access the database and get the view back.

1.6 OUTLINE OF THE BOOK

Chapter 1 describes present day and future Internet architecture, the structure of Points of Presence, where core and edge routers are interconnected with Layer-2 switches. It shows a router architecture, where a large number of line cards are interconnected by a switch fabric. It also includes a router controller that updates the forwarding tables and handles network management. Two commercial, state-of-the-art routers are briefly described. It also outlines the challenges of building a high-speed, high-performance router.

Chapter 2 describes various schemes to look up a route. For a 10-Gbps line, each lookup is required to complete within 40 ns. In present day forwarding tables, there can be as many as 500,000 routes. As more hosts are added to the Internet, the forwarding table will grow one order of magnitude in the near future, especially as IPv6 emerges. Many high-speed lookup algorithms and architectures have been proposed in the past several years and can be generally divided into ternary content address memory (TCAM)-based or algorithmic-based. The latter uses novel data structure and efficient searching methods to look up a route in a memory. It usually requires a larger space than the TCAM approach, but consumes much less power than a TCAM.

Chapter 3 describes various schemes for packet classification. To meet various QoS requirements and security concerns, other fields of a packet header, beyond the IP destination address, are often examined. Various schemes have been proposed and are compared in terms of their classification speed, the capability of accommodating a large number of fields in the packet headers, and the number of filtering rules in the classification table. Because more fields in the packet header need to be examined, in prefix or range formats, it imposes greater challenges to achieve high-speed operations. TCAM is a key component in packet classification, similar to route lookup. Various algorithmic approaches have been investigated by using ordinary memory chips to save power and cost.

Chapter 4 describes several traffic management schemes to achieve various QoS requirements. This chapter starts by explaining Integrated Services (IntServ) and Differentiated Service (DiffServ). Users need to comply with some contract to not send excessive traffic to the network. As a result, there is a need to police or shape users' traffic if they don't comply with the predetermined contract. Several schemes have been proposed to meet various QoS requirements. They are divided into two parts, packet scheduling to meet various delay/bandwidth requirements and buffer management to provide different loss preferences.

Chapter 5 describes the basics of packet switching by showing some fundamental switching concepts and switching fabric structures. Almost all packet switching fabrics can route packets autonomously without an external configuration controller, as circuit switches do. One of the important challenges in building large-scale, high-performance switches is to resolve packet contention, where multiple packets are heading to the same output and only one of them can be transmitted at a time. Buffers are used to temporarily store those packets that lost the contention. However, the placement of the buffers, coupled with contention resolution schemes, determines much of the switch's scalability, operation speed, and performance.

Chapter 6 describes the shared-memory switch, which is the best performance/cost switch architecture. Memory is shared by all inputs and outputs, and thus has the best buffer utilization. In addition, delay performance is also the best because of no head-of-line blocking. On the other hand, the memory needs to operate at the speed of the aggregated bandwidth of all input and output ports. As the line rate or the port number increases, the switch size is limited due to the memory speed constraint. Several architectures have been proposed to tackle the scalability issue by using multiple shared-memory switch modules in parallel. The difference between various proposed ideas lies in the ways of dispatching packets from the input ports to the switch modules.

Chapter 7 describes various packet scheduling schemes for input-buffered switches. The complexity of resolving packet contention can cause the system to bottleneck as the switch size and the line speed increase. The objective is to find a feasible scheduling scheme (e.g., in a time complexity of O(log N) where N is the switch size), close to 100 percent throughput, and low average delay. Several promising schemes have been proposed to achieve 100 percent throughput without speeding up the internal fabric operation speed. However, their time complexity is very high and prohibits them from being implemented for real applications. One practical way to maintain high-throughput and low delay is to increase the internal switch fabric's operation speed, for example, twice the line rate, to compensate for the deficiency of contention resolution schemes. However, it requires output buffers and, thus, increases the implementation cost. Most packets are delivered to the output buffers and wait there. Thus, some kind of backpressure mechanism is needed to throttle the packets from jamming at the output buffers and from being discarded when buffers overflow.

Chapter 8 describes banyan-based switches. They have a regular structure to interconnect many switch modules that can be 2 × 2 or larger. The multistage interconnection network was investigated intensively in early 1970s for interconnecting processors to make a powerful computer. The banyan-based switches received a lot of attention in early 1980s when people started fast packet switching research. One of the reasons is that the switch size can be scaled to very large by adding more stages. However, the interconnection wire can be very long due to the shuffle type of interconnections and they can occupy a large space, inducing considerable propagation delay between the devices, and difficulty in synchronizing the switch modules on the same stage. As a result, one can rarely find a commercial switch/router that is built with the banyan structure.

Chapter 9 describes Knockout switches. It has been proven that output-buffered or shared-memory switches demonstrate the best performance, where packets from all inputs need to be stored in an output buffer if they all are destined for the same output. The memory speed constraint limits the switch size. However, what is the probability that all incoming packets are destined to the same output? If the probability is very low, why do we need to have the output buffer receive all of them at the same time? A group of researchers at Bell Labs in the late 1980s tried to resolve this problem by limiting the number of packets that can arrive at an output port at the same time, thus relaxing the speed requirement of the memory at the output ports. Excessive cells are discarded (or knocked out) by the switch fabric. Various switch architectures using the knockout principle are presented in this chapter.

Chapter 10 describes the Abacus switch that was prototyped at Polytechnic University by the first author of the book. It takes advantage of the knockout principle by feeding back those packets that are knocked out in the first round to retry. As a result, the packets will not be discarded by the switch fabric. The Abacus switch resolves contention by taking advantage of the cross-bar structure of the switch. It can also support the multicasting function due to the nature of the cross-bar structure and the arbitration scheme used in each switch element. The switch fabric has been implemented on ASICs.

Chapter 11 describes crosspoint buffered switches. There are several variants depending where the buffers are placed – only at the crosspoints, both at the inputs and the crosspoints, or at the inputs, the crosspoints, and the outputs. The crosspoint buffer increases performance from input-buffered switches, where performance degradation is due to the head-of-line blocking. In the crosspoint switches, packets are temporarily stored in the crosspoint buffer, allowing multiple packets to be sent out from the same input, which is not possible for the input-buffered switch. However, the trade-off is to implement the memory within the switch fabric. With today's very large scale integration (VLSI) technology, the on-chip memory can be a few tens of megabits, which is sufficient to store a few tens of packets at each crosspoint. Another advantage of this switch architecture is that it allows packet scheduling from inputs to the crosspoint buffers and packet scheduling from the crosspoint buffers to the outputs to be independent, thereby creating the possibility of exploring different scheduling schemes individually to maximize the overall performance.

Chapter 12 describes Clos-network switches. This structure was proposed by Charles Clos for making a scalable telephone circuit switch. It has been used in packet switch fabrics because of its regular interconnections and scalability. Multiple switch modules are interconnected in a Clos-type structure. Each switch module's size is usually determined by the input/output pin count of a chip or the connectors of a printed circuit board. Each switch module can be a crossbar switch with or without buffers. If without buffers, a global contention resolution scheme is required. If with buffers, the contention can be done within the switch modules in a distributed manner. As the switch grows larger and larger, it is more practical to resolve the contention in a distributed manner. However, because of multiple possible paths between each input and each output pair, the challenge becomes choosing a switch module in the center stage. If not carefully chosen, the load is not balanced among the switch modules in the center stage, causing some center modules to be more congested than others. In addition, due to various delays of the multiple paths between an input/output pair, maintaining the order of the packets in the same flow becomes very challenging.

Chapter 13 describes a practical multi-plane, multi-stage buffered switch, called TrueWay. It has the features of scalability, high-speed operations, high-performance, and multicasting. It resolves the issues of: (1) How to efficiently allocate and share the limited on-chip memories; (2) How to intelligently schedule packets on multiple paths while maximizing memory utilization and system performance; (3) How to minimize link congestion and prevent buffer overflow (i.e., stage-to-stage flow control); and (4) How to maintain packet order if the packets are delivered over multiple paths (i.e., port-to-port flow control). A small-scale TrueWay switch has been prototyped using field programmable gate array (FPGA) and SERDES chips with signals running at 3.125 Gbps at the backplane.

Chapter 14 describes load-balanced switches. The idea of this type of switch is very interesting. Packets from each input are temporarily and evenly distributed to all other output ports. They are then forwarded to their final destinations. By doing this, the switch fabrics can operate in a cyclic-shifting configuration at each time slot and still achieve 100 percent throughput. One challenging issue of this kind of switch is to maintain packet order. The difference of several proposed schemes lies in the way of resolving the packet out-of-order issue. Since the packets traverse from the input to a temporary output and then from that port to the final output port, one either needs to use two separate switch fabrics or have one switch fabric running at twice the speed of the input. People will argue that the 100 percent throughput is achieved because of the speedup factor of 2. However, the biggest incentive for this kind of switch is that there is no need for an arbitrator to resolve packet contention.

Chapter 15 describes optical packet switches. 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 and all-optical packet switches. In either case, contention resolution among the arriving packets is handled electronically. It is very challenging to store contending packets in an optical buffer, which is usually implemented by an optical delay line. The storage size and the time that a packet can be stored are quite limited when using optical delay lines. Another challenging issue is to align the optical packets before storing them on the optical delay lines. It requires tremendous effort to align them when traversing different distances from different sources. Until there is a major breakthrough in optical buffering technology, it will remain very difficult to implement all-optical packet switches.

Chapter 16 describes high-speed router chip sets. This is the most unique section of the book. Up to this point, emphasis has been on learning the background information necessary to implement a high-performance router, including IP address lookup, packet classification, traffic management, and various techniques to build large-scale, high-speed switch fabrics. This section describes practical commercial chips that are used to implement all the above. Thus, it paves a way for tightly combining theory with practice. These chips include: (1) Network processors for flexible packet processing; (2) Co-processors for route lookup and packet classification; (3) Traffic managers; and (4) Switch fabrics.

REFERENCES

[1] G. Vareille, F. Pitel, and J. F. Marcerou, '3-Tbit/s (300 × 11.6 Gbit/s) transmission over 7380 km using C + L band with 25 GHz channel spacing and NRZ format,' in *Proc. Optical Fiber Communication Conference and Exhibit*, Anaheim, California, vol. 4, pp. PD22-1-3 (Mar. 2001).

[2] R. Ryf *et al.*, '1296-port MEMS transparent optical crossconnect with 2.07 petabit/s switch capacity,' in *Proc. Optical Fiber Communication Conference and Exhibit*, Anaheim, California, vol. 4, pp. PD28-P1 (Mar. 2001).

[3] N. Spring, R. Mahajan, and D. Wetherall, 'Measuring ISP topologies with rocketfuel,' in *Proc. ACM SIGCOMM*, Pittsburgh, Pennsylvania, pp. 133–145 (Aug. 2002).

[4] N. Mckeown, 'A fast switched backplane for a gigabit switched router,' *Business Communications Review*, vol. 27, no. 12 (Dec. 1997).

[5] C. Partridge *et al.*, 'A fifty gigabit per second IP router,' *IEEE/ACM Transactions on Networking*, vol. 6, no. 3. pp. 237–248 (June 1998).

[6] *VIT10: 10G transponder and VSC8173/75: physical layer multiplexer/demultiplexer*, Vitesse. [Online]. Available at: http://www.vitesse.com

[7] *CA16: 2.5 Gbit/s DWDM with 16-channel 155 Mb/s multiplexer and demultiplexer and TB64: Uncoded 10 Gbit/s transponder with 16-channel 622 Mbit/s Multiplexer and Demultiplexer*, Agere. [Online]. Available at: http://www.agere.com

[8] M. C. Chow, *Understanding SONET/SDH: Standards and Applications*, Andan Publisher, 1995.

[9] A. Romanow and R. Oskouy, 'A performance enhancement for packetized ABR and VBR + data,' *ATM Forum 94-0295*, Mar. 1994. [Online]. Available at: http://www.mfaforum.org

[10] S. Floyd and V. Jacobson, 'Random early detection gateways for congestion avoidance,' *IEEE/ACM Transactions on Networking*, vol. 2, no. 4, pp. 397–413 (Aug. 1993).

[11] *Quality of Service (QoS) Networking*, Cisco System, June 1999, white paper.

[12] G. Armitage and K. Adams, 'Package reassembly during cell loss,' *IEEE Network*, vol. 7, no. 5, pp. 26–34 (Sept. 1993).

[13] D. Stiliadis and A. Varma, 'A general methodology for design efficient traffic scheduling and shaping algortihms,' in *Proc. IEEE INFOCOM'97*, Kobe, Japan, pp. 326–335 (Apr. 1997).

[14] J. C. R. Bennett and H. Zhang, 'Hierarchical packet fair queuing algorithms,' *IEEE/ACM Transactions on Networking*, vol. 5, no. 5, pp. 675–689 (Oct. 1997).

[15] H. J. Chao and X. Guo, *Quality of Service Control in High-Speed Networks*, John Wiley & Sons, Inc., Sept. 2001.

[16] 'Juniper networks t-series core platforms,' Juniper Inc. [Online]. Available at: http://www.juniper.net/products/tseries/

[17] R. Sudan and W. Mukai, *Introduction to the Cisco CRS-1 Carrier Routing System*, Cisco Systems, Inc, Jan. 1994.

[18] *Plugable fiber optic link* (12 × 2.5 *Gbps*), Paracer. [Online]. Available at: http://www.paracer.com

[19] *M21150 and M21155 144 × 144 3.2 Gbps Crosspoint Switch*, Mindspeed. [Online]. Available at: http://www.mindspeed.com

[20] *VC3003 140 × 140 Multi-Rate Crosspoint Switch with Clock, Data Recovery at Each Serial Input*, Velio. [Online]. Available at: http://www.velio.com

[21] M. Gupta and S. Singh, 'Greening of the Internet,' in *Proc. ACM SIGCOMM03*, Karlsruhe, Germany, pp. 19–26 (Aug. 2003).

[22] 'ISO 9596, information processing systems, open system interconnection, management information protocol specification, common management information protocol.' ISO, Nov. 1990.

[23] 'ISO 7498, information processing systems, open system interconnection, basic reference model part 4, OSI management framework.' ISO, Oct. 1986.

[24] 'ISO/IEC DIS 10165-1, ISO/IEC DIS 10165-2, ISO/IEC DIS 10165-4, information processing systems, open system interconnection, structure of management information.' ISO, July 1990.

[25] J. Case, R. Mundy, D. Partain, and B. Stewart, *Introduction to Version 3 of the Internet-standard Network Management Framework*, RFC 2570 (Informational), Apr. 1999, obsoleted by RFC 3410. [Online]. Available at: http://www.ietf.org/rfc/rfc2570.txt

[26] U. Black, *Network Management Standards: SNMP, CMIP, TMN, MIBs, and Object Libraries*, 2nd ed., McGraw-Hill, 1994.

[27] W. Stallings, *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*, Addison-Wesley, Massachusetts 1999.

[28] *SNMPv1, SNMPv2, and SNMPv3*, SNMP Research International. [Online]. Available at: http://www.snmp.com

[29] *MIB compiler, MIB browser, and SNMP APIs*, AdventNet, Inc. [Online]. Available at: http://www.adventnet.com/products/javaagent/snmp-agent-mibcompiler.html