Chapter 1

Multiservice Loss Systems

A loss system is a collection of resources to which calls, each with an associated *holding time* and *class*, arrive at random instances (see Figure 1.1). An arriving call either is admitted into the system or is blocked and lost; if the call is admitted, it remains in the system for the duration of its holding time. The admittance decision is based on the call's class and the system's state.

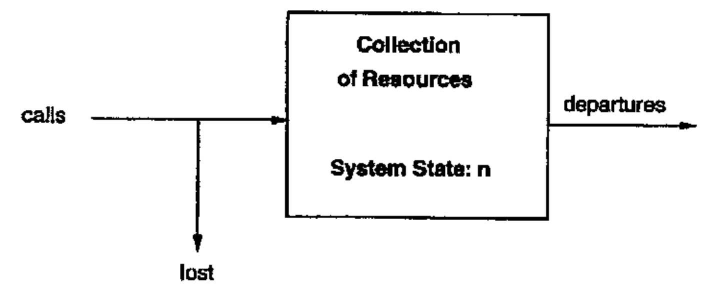


Figure 1.1: The generic loss system.

A loss system is fundamentally different from a queueing system because a call's system sojourn time is equal to its holding time. In this chapter we informally discuss how loss systems can model a variety of telecommunication technologies; in the subsequent chapters we elaborate on these models and applications.

1.1 The Erlang Loss System

The Erlang loss system is the simplest of all loss systems, consisting of a link of C circuits to which calls of one class arrive. Each call in progress occupies one of the circuits, and an arriving call is blocked when the system is full. The calls arrive according to a Poisson process with rate λ, and the call holding times are independent and identically distributed with mean $1/\mu$. The fraction of calls blocked, B, is given by the celebrated Erlang loss formula,

$$B = \frac{\rho^C/C!}{\sum_{c=0}^{C} \rho^c/c!},$$

where $\rho := \lambda/\mu$. In 1917 A.K. Erlang stated and proved this result for the case of exponentially distributed holding times.

Since its discovery, the Erlang loss formula has had a profound impact on the design of telecommunication networks. Up until the 1980s, when personal computers became widely available, Erlang loss tables could be found on the desk of just about every engineer designing telephone networks. Engineers used the tables on a daily basis to determine the minimum number of circuits, C, to meet a specified level of blocking performance, B_{max}. Today, the formula is implemented in computers for a broad range of applications. It is combined with spreadsheet programs so that telecommunication sales personnel can price private access lines for their customers. And it is often a critical subroutine in complex software packages which aid engineers to design and dimension long-distance networks with dynamic routing.

Because the Erlang loss formula is so relevant to the design of telephone networks, academic and industrial researchers have studied it in great depth. Syski gives a comprehensive treatment in his classic book [151]. We shall frequently refer to the formula while studying networks and multiservice systems, not only because it is an excellent springboard for discussing these more complex models, but also because it often serves as a core subroutine for their algorithmic analysis.

1.2 Loss Networks with Fixed Routing

Whereas the Erlang loss system sheds great insight on the performance of a single link, a loss network can accurately model an entire telephone network consisting of multiple links and switches. We shall need to distinguish between loss networks with fixed routing and loss networks with dynamic routing. With fixed routing, an arriving call requests establishment on a specific route; if sufficient resources are not available along the route, the call is blocked. With dynamic routing, if the first-choice route is not available, other routes may be tried.

Postponing the formal definition of a loss network until Chapter 5, we now show how a loss network with fixed routing can model the private voice network of a company. This company has four offices scattered over a metropolitan region. Each office has a private branch exchange (PBX), which enables the employees of the same office to call each other without having to access a public network. The company must nevertheless use the infrastructure of the public network when the employees of one office wish to communicate with the employees of another. To allow for the interoffice traffic, the company interconnects its PBXs by leasing links from a public telephone company, where each link consists of a finite number of circuits. As shown in Figure 1.2, these leased links connect the PBXs to a centrally located switch, also owned by the public telephone company.

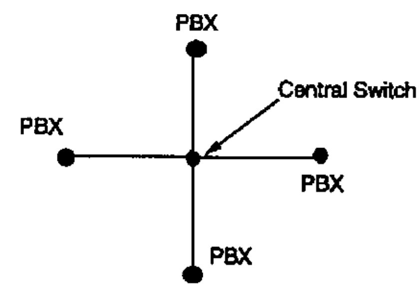


Figure 1.2: A telephone network with a star topology.

To model this telephone network as a loss system, we need to define the *class* of a call and the *state* of the network. A call's class is simply its route; since there are four PBXs in this example, there are six routes and hence six classes. The state of the network is a vector $\mathbf{n} = (n_1, \ldots, n_6)$, where n_k is the number of class-k calls in progress. Since each call occupies one circuit in each of the links along its route, the number of possible states is finite; denote $\mathcal{S}$ for the set of all states. Also denote $\mathcal{S}_k$ for the set of states with room for another class-k call. Specifically, $\mathbf{n} \in \mathcal{S}_k$ if and only if $\mathbf{n} + \mathbf{e}_k \in \mathcal{S}$, where $\mathbf{e}_k$ is the vector of all zeros except for a one in the kth component.

As for the Erlang loss system, there is an explicit formula for blocking probability for a loss network with fixed routing. The formula hinges on two minor assumptions. The first is that the call arrival processes for the six classes are independent and Poisson; let λ_k denote the arrival rate for class-k calls. The second assumption is that the call holding times are independent of each other, independent of the arrival processes, and for each class have an identical distribution; let $1/\mu_k$ denote the average holding time of a class-k call and let $\rho_k := \lambda_k/\mu_k$. With these assumptions, we shall see in Chapter 5 that the probability of blocking a class-k call is

$$B_k = 1 - \frac{G_k}{G},$$

where G and G_k are *normalization constants*, defined by

$$G := \sum_{\mathbf{n} \in \mathcal{S}} \prod_{k=1}^{6} \frac{\rho_k^{n_k}}{n_k!}$$

and

$$G_k := \sum_{\mathbf{n} \in \mathcal{S}_k} \prod_{l=1}^{6} \frac{\rho_l^{n_l}}{n_l!}.$$

Engineers can use this result to dimension the capacity of the links, minimizing monthly leasing charges while meeting performance requirements for interoffice blocking.

Although these formulas are remarkably explicit, it is a non-trivial problem to calculate the sum in the normalization constant because the sum has many terms for just moderate link capacities. Developing efficient methods for this calculation is one of the major projects of

this book. Two methods are explored in great detail. The first uses *convolution algorithms* to perform the requisite sums and products in a specific order. These algorithms are not efficient for all topologies, but they do perform well for access networks and hierarchical access networks — network topologies which serve an important role for local and long-distance telephone companies. The second method, studied in Chapter 6, employs *Monte Carlo summation* to estimate the normalization constant. It applies to arbitrary topologies, but gives confidence intervals instead of exact results.

We also explore two other approaches for assessing blocking performance. The first is to upper bound the blocking probabilities by way of the *product bound.* The second is to approximate blocking probabilities with the *reduced load approximation* (also referred to as the Erlang fixed-point approximation) and solve the associated fixed-point equation with repeated substitutions.

To dimension the network and optimize its performance, the engineer needs to understand how increases in call volumes impact long-term revenue. Let each class-k call in progress generate revenue at rate r_k dollars per second, and let W denote the long-run average revenue generated by the interoffice calls. The engineer would like to know the rate of change of long-run average revenue with respect to arrival rates. We shall see that this revenue sensitivity can be expressed as

$$\frac{\partial W}{\partial \lambda_k} = (1 - B_k)\left(\frac{r_k}{\mu_k} - c_k\right),$$

where c_k is the implied cost of class-k calls, that is, the loss in revenue due to additional blocking when inserting a new class-k call in the network in equilibrium. This expression for revenue sensitivity has an intuitive interpretation: Increasing λ_k by a small amount will cause additional class-k calls to arrive infrequently; an additional call is admitted with probability $1-B_k$ and, if admitted, contributes an expected revenue of r_k/μ_k at the expected cost of c_k.

In Chapters 5 and 6 we shall address methods to approximate and exactly calculate revenue sensitivities. Both the reduced load approximation and Monte Carlo summation will play an important role.

1.3 Loss Networks with Dynamic Routing

Many long-distance telephone companies provide dynamic routing in their networks. The current trend worldwide is to implement dynamic routing over (logically) fully connected networks, for which each node pair has a direct route and a number of two-link alternative routes (see Figure 1.3). The modern routing schemes for these networks first attempt to establish a new call on its direct route; if the direct route is not available, they either establish the call on a two-link alternative route or block the call. The decision whether to block the call or establish it on a particular alternative route depends on the specific routing scheme and the state of the network.

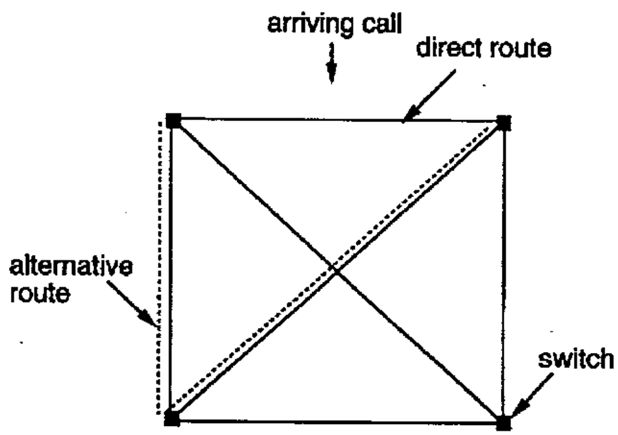


Figure 1.3: A fully connected network.

From a historical perspective, one of the most important routing schemes is Dynamic Nonhierarchical Routing (DNHR), AT&T's routing scheme in the late 1980s for its domestic network. DNHR searches through a fixed sequence of alternative routes until a free end-to-end circuit is found. In the early 1990s, AT&T replaced DNHR with Real-Time Network Routing (RTNR), which, in essence, selects the alternative route that has the largest number of end-to-end free circuits; this routing scheme is also referred to as *least loaded routing*. Other state-dependent routing schemes include Dynamic Alternative Routing, planned for British Telecom's domestic telephone network, Dynamically Controlled Routing, developed by Bell Northern Research and implemented in the Trans Canadian Network, and Forward Looking Routing, developed at Bellcore and used in trials in some local telephone networks in the United States.

Although networks with dynamic routing fail to have a closed-form expression for their blocking probabilities, they are still amenable to analysis. In Chapter 7 we shall explore two analytical techniques. The first is to bound long-run average revenue by way of *max-flow bounds*. The second is to approximate blocking probabilities with a reduced load approximation.

1.4 The ATM Multiplexer

Up to this point our examples have all been *single-service* loss systems, that is, systems for which a call occupies exactly one circuit in each link along its route. But the focus of this book is on *multiservice* loss systems — networks whose calls have heterogeneous bandwidth requirements. The simplest telecommunication example of a multiservice loss system is an asynchronous transfer mode (ATM) multiplexer. We shall explore the connection performance of the ATM multiplexer in Chapters 2, 3, and 4. We present here an overview of some of the results in those chapters.

We first give some ATM terminology. A *source* is a terminal such as a telephone handset, a video player, or a multimedia computer. When a source wants to transmit information, it requests establishment of a *virtual channel* (VC). Once a source has established a VC, it generates a stream of *cells*, each cell consisting of 53 bytes. With a slight abuse of language, we shall write "an established VC" or sometimes more simply "a VC" for "a source with an established VC". A typical cell stream generated by an established VC consists of silent periods, during which no cells are generated, and activity periods, during which cells are generated at the *peak rate*. An *ATM multiplexer* is a buffer and a high-speed link; the buffer receives the cells generated by established VCs and transmits these cells, one after another, onto the high-speed link. Assume that VCs belong to a finite set of *services* (see Figure 1.4). Examples of services include voice (that is, an ordinary telephone call), low- and high-quality facsimile, video conference, video on demand, file transfer, image retrieval, and LAN interconnect.

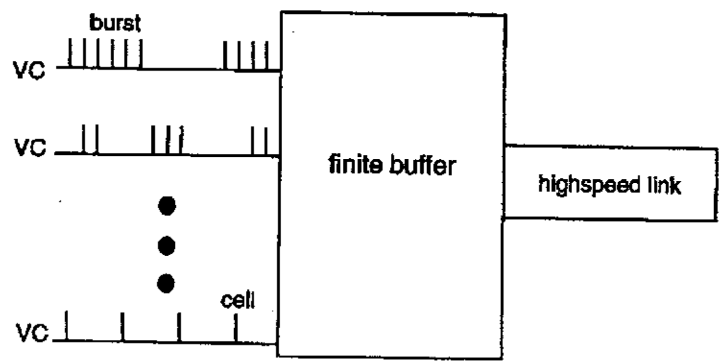


Figure 1.4: An ATM multiplexer.

During periods when the aggregate cell arrival rate exceeds the link capacity, the multiplexer can significantly delay or even lose cells. A VC's allowable cell delay and loss are specified by its quality of service (QoS) requirements; for example, the QoS requirement for a VC transporting a voice service might be that the fraction of lost cells be less than 10^{-6}. To guarantee that all established VCs meet their QoS requirements, the multiplexer may have to deny certain VC establishment requests — thus the need for an *admission policy*. An admission policy is said to meet the QoS requirements if all established VCs meet their QoS requirements when the policy is applied.

Below we briefly discuss three admission policies. In the ensuing chapters we shall investigate them in greater detail as well as introduce other admission policies.

Admission Based on Peak Rates

Let C denote the transmission capacity of the high-speed link, K denote the number of services, and $b_1, \ldots, b_K$ denote the peak rates for the K services. The *VC profile* is $(n_1, \ldots, n_K)$, where n_k is the number of class-k VCs in progress. Since VCs arrive and depart, the VC profile changes with time.

Peak-rate admission admits a new service-k VC if and only if

$$b_k + \sum_{l=1}^{K} b_l n_l \leq C,$$

where $(n_1, \ldots, n_K)$ is the current VC profile. This condition ensures that cells experience negligible delay and no loss at the buffer; consequently, the QoS requirements are met. It is important to note that an ATM multiplexer with peak-rate admission is, with regard to VC dynamics, a loss system: a call is a VC, the class of a VC is its service type, and the state of the loss system is the VC profile. We refer to this loss system as the *stochastic knapsack*.

It is straightforward to determine blocking probabilities for the stochastic knapsack if VCs arrive according to a Poisson process. Let λ_k and $1/\mu_k$ denote the arrival rate and mean holding time of service-k VCs, and let $\rho_k := \lambda_k/\mu_k$. Let $\mathcal{S}$ denote the set of all VC profiles, that is,

$$\mathcal{S} := \{(n_1, \ldots, n_K) \; : \; \sum_{k=1}^{K} b_k n_k \leq C \}.$$

Let $\mathcal{S}_k$ denote the set of all VC profiles that have room for another service-k VC, that is,

$$\mathcal{S}_k := \{(n_1, \ldots, n_K) \; : \; \sum_{l=1}^{K} b_l n_l \leq C - b_k \}.$$

We shall see that the probability of blocking a service-k VC is

$$B_k = 1 - \frac{G_k}{G},$$

where G and G_k are the normalization constants:

$$G := \sum_{\mathbf{n}\in\mathcal{S}} \prod_{k=1}^{K} \frac{\rho_k^{n_k}}{n_k!}$$

and

$$G_k := \sum_{\mathbf{n}\in\mathcal{S}_k} \prod_{l=1}^{K} \frac{\rho_l^{n_l}}{n_l!}.$$

Revenue sensitivity can also be expressed as a function of normalization constants. Owing to the special structure of the sets $\mathcal{S}$ and $\mathcal{S}_k$, a simple and efficient recursive algorithm can rapidly calculate the normalization constants and, consequently, the blocking probabilities and revenue sensitivities.

Since it accurately models many key features of the ATM multiplexer, the stochastic knapsack will be studied in depth and from a variety of perspectives. One such perspective is that of the behavior of its blocking probabilities when arrival rates are increased. We shall see that this *monotonicity behavior* is quite complex — an increase in the arrival rate for a particular service can either favorably or adversely affect blocking probabilities. Nevertheless, we shall obtain monotonicity results which shed significant light on the qualitative structure of the stochastic knapsack.

Primarily due to the marvelous advances in fiber-optic communications, transmission capacity has been increasing at a rapid pace. Commensurate with this growth is a demand for more bandwidth — new services such as video on demand and multimedia have a thirst for bandwidth that seems impossible to quench. These increases in demand and capacity motivate us to study the stochastic knapsack from another perspective, that of its asymptotic behavior as capacity and demand approach infinity. The asymptotic analysis will lead to several fascinating results. To give the flavor of an asymptotic result, let the transmission capacity and the offered traffic be large, and suppose that they are roughly equal, that is, we suppose

$$\sum_{k=1}^{K} \frac{b_k \lambda_k}{\mu_k} \approx C.$$

Then we shall see that

$$B_k \approx \frac{b_k \delta}{\sqrt{C}},$$

where δ is a constant that is independent of k and C. This result implies that when the transmission capacity is large and nearly equal to the offered traffic, blocking probability for a service is roughly proportional to the service's peak rate. It also implies that blocking probabilities decay at a rate of $1/\sqrt{C}$.

Admission Based on Effective Bandwidths

The multiplexer operates in the *statistical multiplexing mode* if the admission policy permits VC profiles whose aggregate peak rates exceed the transmission capacity of the link, that is, if the policy permits a VC profile $(n_1, \ldots, n_K)$ such that

$$\sum_{k=1}^{K} b_k n_k > C.$$

During a period of time when the above inequality holds, the multiplexer can lose and significantly delay cells.

For statistical multiplexing, the performance of a specific admission policy is characterized by two types of measures: *cell performance* and *connection performance*. Cell performance is the delay and loss due to cell accumulation and overflow at the buffer. Connection performance is the rejection probability of VC establishment requests. There is a clear tradeoff between cell and connection performance: If we admit (respectively reject) more VCs, the buffer will become more (respectively less) congested with cells.

Frequently discussed in the ATM literature, *effective-bandwidth admission* is an admission policy which is easy to implement. This admission policy is characterized by a vector $(b_1^e, \ldots, b_K^e)$ and admits a new service-k VC if and only if

$$b_k^e + \sum_{l=1}^{K} b_l^e n_l \le C,$$

where $(n_1, \ldots, n_K)$ is the current VC profile. What are appropriate values for $b_1^e, \ldots, b_K^e$, the effective bandwidths of the K services? If

$b_k^e = b_k$ for all services, then the policy reduces to peak-rate admission and, consequently, the QoS requirements are met. If the b_k^e's are small relative to the b_k's, then the policy admits more VCs than does peak-rate admission but may no longer respect the QoS requirements. But no matter what the choice for the effective bandwidths, from the viewpoint of the VC dynamics, the ATM multiplexer with effective-bandwidth admission is accurately modeled by the stochastic knapsack.

Admission Based on Service Separation

This admission/scheduling policy allows for a degree of statistical multiplexing yet ensures satisfaction of the QoS requirements. To define it, first consider a multiplexer of buffer capacity A which multiplexes n permanent service-k VCs but no VCs from services other than k. Denote $\beta_k(n)$ for the minimum amount of transmission capacity needed for the QoS service requirements to be met for the n VCs. Since $\beta_k(\cdot)$ is a function of a single parameter, n, it should not be difficult to determine by a simulation or analytical analysis of the cell dynamics. We call this function the service-k capacity function.

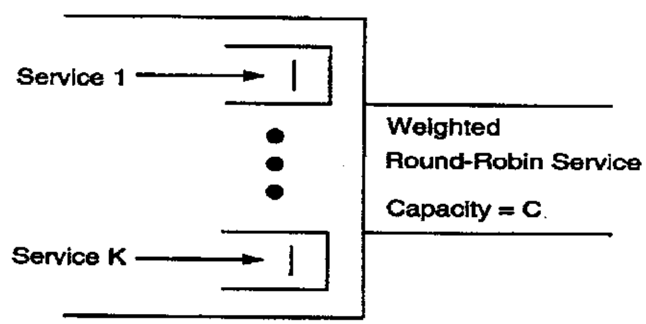


Figure 1.5: A multiplexer implementing service separation.

Returning to the original multiservice ATM multiplexer, partition the buffer into K mini-buffers, each of size A. Dedicate the kth mini-buffer to service-k cells; see Figure 1.5. When the VC profile is $(n_1, \ldots, n_K)$, require the link to serve the K mini-buffers with

a weighted round-robin discipline, with mini-buffer k served at rate $\beta_k(n_k)$. *Service separation* admits a new service-k VC if and only if

$$\beta_1(n_1) + \cdots + \beta_k(n_k + 1) + \cdots \beta_K(n_K) \le C.$$

This policy, discussed in greater detail in Chapter 4, satisfies the QoS requirements and blocks significantly fewer VCs than does peak-rate admission. Although its VC profile space,

$$\mathcal{S} = \{(n_1, \ldots, n_K) \ : \ \beta_1(n_1) + \cdots + \beta_K(n_K) \le C\},$$

is not the VC profile space of the stochastic knapsack, we can still efficiently calculate VC blocking probabilities with convolution and Monte Carlo algorithms. Observe that by setting $\beta_k(n) = b_k^e n$ for all services, this scheme becomes effective bandwidth admission.

1.5 ATM Networks

Although tractable mathematically, loss models for the connection performance of ATM networks are surprisingly subtle and intricate, owing to the complex interactions among admission, scheduling, and routing. In this section we explore some of the important subtleties with a simple example.

An ATM network is depicted in Figure 1.6. It transports two services, labeled service 1 and service 2, and has two routes, the top route and the bottom route. The top route starts at the top source switch, passes through the intermediate switch, and ends at the destination switch. The bottom route is the same except it starts at the bottom switch. For a network, a VC is now a virtual connection for a service-route pair; since there are four service-route pairs, there are four classes of VCs. For a network, the QoS requirements are end-to-end; for example, the QoS requirements for a service-k VC along the top route might be that the fraction of its cells lost at the top source switch plus the fraction of its cells lost at the intermediate switch be less than ϵ_k.

We can design an ATM network with a wide variety of admission/scheduling policies, each engendering a different degree of service

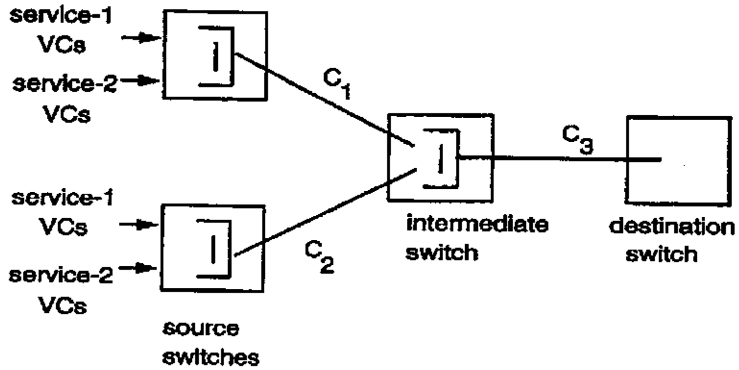
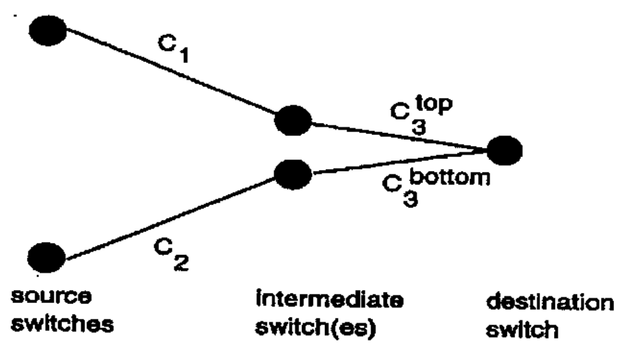


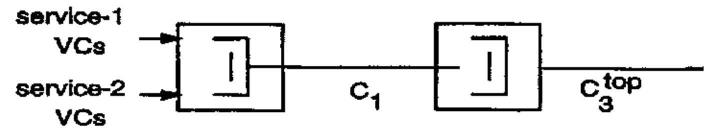
Figure 1.6: A simple topology for an ATM network.

and route separation. To give an example, for the network in Figure 1.6 we now discuss one such policy — *dynamic-service/static-route separation.*

First we describe how the policy statically separates the routes. Partition the buffer in the intermediate switch into two mini-buffers, one for each route. At this switch, require the cells of a route to be directed to the corresponding mini-buffer. Let the two mini-buffers be served at rates C_3^{top} and C_3^{bottom}, where $C_3^{top} + C_3^{bottom} \leq C$. These modifications transform the intermediate switch into two separate switches, giving a topology with two separate routes:



Having separated the routes, we can analyze each one in isolation. Let us focus on the top route:



For this isolated network, we now describe how the policy dynamically separates the services. At both the source and intermediate switches, partition the buffer into two mini-buffers, one for each service; for simplicity, assume all mini-buffers have buffer capacity A. Let $\beta_k(n)$ be the capacity function for service-k, as defined in the preceding section. Let (n_1, n_2) denote the current VC profile, where n_k is the number of service-k VCs in progress (on the top route). At both the source and intermediate switches, require the mini-buffers for the kth service to be served at rate $\beta_k(n_k)$. Finally, let $C^{top} = \min(C_1, C_3^{top})$.

Dynamic-service/static-route separation admits a new service-1 VC to the top route if and only if

$$\beta_1(n_1 + 1) + \beta_2(n_2) \leq C^{top}.$$

The policy for the other service-route pairs is defined in an analogous manner. We shall argue that this admission/scheduling policy ensures that the QoS requirements are met end-to-end.

Dynamic-service/static-routing separation is just one of many interesting admission/scheduling policies that we investigate in Chapter 6. Most of these policies have an explicit expression for their blocking probabilities, and are therefore amenable to analysis by Monte Carlo summation. They can also be analyzed with reduced load approximations. In Chapter 8 we shall also study these policies in the context of ATM networks with dynamic routing.

1.6 Multiservice Interconnection Networks

Microelectronic chip considerations typically dictate that the switch fabric, the heart of the ATM switch, reside on a single board or even on a single chip. This in turn limits the number of input and output ports for the switch to some small value. But large ATM switches in public ATM networks typically require a larger number of input and output ports. In order to build these larger switches, switch designers must interconnect a number of small switches, referred to as modules.

By interconnecting a sufficient number of modules, a large switch can be built with any number of input and output ports. But these interconnections may introduce undesirable blocking within the interconnection network. In Chapter 9 we explore how interconnection networks can be designed with minimum complexity so that VC switch blocking is eliminated. We shall consider both strictly nonblocking and rearrangeable interconnection networks.