Efficiently Providing Multiple Grades of Service with Protection Against Overloads in Shared Resources

Gagan L. Choudhury
Kin K. Leung
Ward Whitt

Multiservice telecommunications systems require different grades of service for different customers, including protection against overloads caused by other customers. One way to provide multiple grades of service, including overload protection, is by partial sharing using upper-limit (UL) bounds, which specify an upper limit on the number of requests a customer is allowed at any time, and guaranteed-minimum (GM) bounds, which guarantee that there will always be space for a minimum number of requests from that customer. These bounds achieve effective separation with sharing and can be efficiently enforced and analyzed. Analysis is made possible by a new algorithm for computing blocking probabilities based on numerical transform inversion.

Resource-Sharing Problems

An important feature of modern multimedia telecommunications systems is the need to simultaneously satisfy demands from different customers who have very different quality and bandwidth requirements. A fundamental problem is to find an effective way to:

- Provide appropriate grades of service to different customers,
- Protect against overloads caused by other customers, and
- Guarantee performance under normal loads.

Some customers will require high-quality service—even in the presence of significant overloads by other customers—whereas other customers might accept occasional degradation in service—provided that the service is available at a lower cost.

This paper addresses the multiservice problem by considering a mathematical resource-sharing model, which is a generalization of the classical Erlang loss model.[1-11] The resource-sharing model includes multiple *resources*, each containing multiple *resource units* that provide service to multiple *customers*. Each customer is a source of a series of *requests*, requiring a number of resource units from each resource. This number of units could vary from customer to customer, and from resource to resource.

If all network requirements are met when a new request arrives, then the system accepts the request, and all the required resource units are held throughout the request holding time. Otherwise, the request is not admitted, and is said to be blocked. (The mathematical model is described in more detail in Panel 2.)

Circuit-Switched Networks. In a standard application to a circuit-switched telecommunications network, the resources are *links*, and the resource units are the *circuits* on these links. For example, each circuit might be a 64-kb/s line. The customers subscribe to different *classes* of services, such as voice, data, and video, and service requests are made via dial-up *calls*. These calls will require one—or more—such circuits simultaneously (often called N x 64 kb/s service) on several different links, depending on the service and origin and destination of the call. Thus, customer needs may differ because of the type and grade of service requested, and the routes required through the network.

A simple circuit-switched telecommunications network is depicted in Figure 1. This figure shows 5 nodes (A through E) and

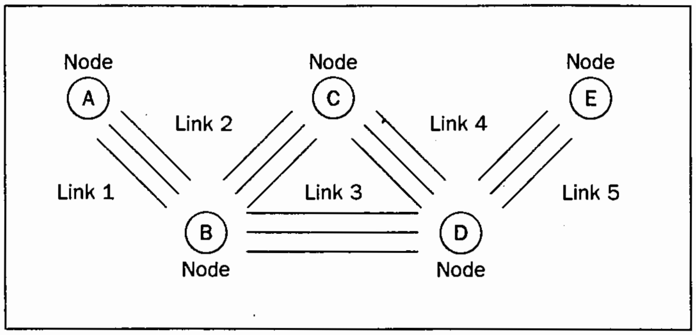

Figure 1. In a typical loss network, the critical resources are the links if the network is circuit-switched, and both nodes and links if the network is packet-switched.

Table I. Example of Network Routes and Links.

Route ID	1	2	3	4	5	6
Link ID	1	2	1, 2	3, 5	4, 5	1, 3, 5

5 links (1 through 5). (In a circuit-switched network, the switches generally are sized to handle all traffic offered by the links; thus we are not concerned here with any blocking problems posed by the switch.) This network serves multiple customers, with the customer being characterized partly by the set of links or *route* that the customer requires. For example, in the example of Figure 1, there might be six routes requiring the subsets of links shown in Table I.

Certain calls, such as dial-up video teleconferencing, may require up to six circuits on each link, while standard voice calls require only one circuit per link. Let's also assume that the network customers have subscribed to one of two classes of service, video conferencing or voice only (which can include low-speed data and facsimile) on a particular route.

The customers on Routes 1, 3, and 6 all use Link 1, and, thus, share the available circuits on that link. When a video call is attempted on Route 6, it requires six circuits each on Links 1, 3, and 5. If, at the time of the attempt, sufficient free circuits are not available on any of these links, then the call is blocked. Unless special measures are taken, the video calls on Route 6 will clearly experience much higher blocking than the voice calls on Route 1.

Or several video conferencing customers could block many subsequent customers from the network because of the high number of circuits and links they occupy, blocking even the single-circuit voice calls. Thus, the network provider, in addition to providing the classes of service customers need, will also want to provide each class of customer protection against overloads from other customers.

Packet-Switched Networks. In a broadband integrated-services digital network (B-ISDN) supported by asynchronous transfer mode (ATM) technology, messages are broken up into small packets called cells. The resources are both nodal *switches* and network *transmission facilities*, and the resource capacity (units) may be the *bandwidth* available at these network facilities. Therefore, both the ATM nodes and links are regarded as critical resources. Figure 1 can also apply to this B-ISDN example.

In contrast to a circuit-switched network, where the request is a single dialup call to establish a dedicated pathway and bandwidth, a B-ISDN customer submits *multiple* requests for service during a connection, where the bandwidth may vary over time. The customers' requests could be viewed at three levels:

- At the micro level, where each ATM cell to be transmitted is regarded as a request;
- At a higher level, where each burst of ATM cells is regarded as a request; or
- At the macro level, where the request is regarded as the total required effective bandwidths associated with all bursts of ATM cells within an established connection.[7,8]

Because of this concept of multiple requests per connection, the probability of a customer experiencing blocking in a shared ATM network could be much higher than in a circuit-switched network—unless the system is properly engineered to support a variety of different traffic characteristics and a variety of different bandwidth requirements.

Fundamental Problems. In any system of shared resources, the resource provider is faced with two fundamental problems:

- The first is the obvious fact that resource units are expensive to provide, so that it is important to have a network designed with no more capacity than is necessary.
- The second is the probabilistic nature of the problem. The submission of customer requests and their hold-

Panel 1. Abbreviations, Acronyms, and Terms

ATM — asynchronous transfer mode
B-ISDN — broadband integrated services digital network
br — blocking ratio parameter, the ratio of the conditional blocking requirement to the nominal blocking requirement.
CP — complete partitioning
CS — complete sharing
customer class — category determined by service, resource requirement, and possibly quality of service
GM — guaranteed minimum
L — guaranteed-minimum bound
t — trunk reservation threshold (see Table II)
TR — trunk reservation
UL — upper limit
x — conditional overload parameter, the percentage of overload in the conditional blocking requirement

ing times are uncertain events that fluctuate over time, so the actual requirements of the customers cannot be known in advance.

However, the pattern of customer requests and request holding times can be predicted in a probabilistic sense. Indeed, it is well known that probabilistic models can be used to characterize customer requirements.

In this uncertain environment, with limited resources, some blocking of customer requests becomes inevitable. It is thus customary to characterize the quality of service received in terms of a customer's *request blocking probability*, that is, the long-run proportion of requests that are blocked in a particular operating regime. If a request blocking probability is too high, then it fails to satisfy a customer, who naturally wants the request blocking probability to be suitably low. On the other hand, if a request blocking probability is too low, then more capacity may have been provided than is needed, a cost factor for the service vendor.

Upper-Limit and Guaranteed-Minimum Bounds

With different kinds of customers, it is often very important to protect each customer from the overloads of the others. If all customers are allowed full access to the resource, one or more customers could actually submit requests at rates higher than their negotiated rates, which can cause other customers to experience unacceptably high blocking probabilities.

One way to protect customers from overloads is to partition the resources into separate portions dedicated to each customer, but such a partitioning tends to be inefficient, because the benefits of sharing are lost, a concern which we will discuss later. However, there are ways, without partitioning, to provide different grades of service to customers sharing a resource—including protection against overloads caused by other customers.

Trunk Reservation. One commonly used control scheme is *trunk reservation (TR)*, which depends on a reservation parameter for each resource and customer. A *new* customer request is admitted if, after admission of the new request, the remaining *free capacity* of each network resource would be greater than or equal to some specified threshold, for example, five percent of the total number of resource units. If the new request for resources, when added to the existing customer resource demands, would leave less than the threshold, then the request would be blocked. The percentage parameter would be different for different customers.

Upper-Limit and Guaranteed-Minimum Bounds. We propose another method to protect against overloads—each customer is assigned *upper-limit (UL)* and *guaranteed-minimum (GM) bounds* on the number of requests that can be in the system simultaneously at any time. The UL bound puts an upper limit on the number of requests from that customer that can be in service at any time. The GM bound guarantees that there will always be resource units available to serve a specified minimum number of requests simultaneously from that customer. The UL bound limits the possible overload from that customer, while the GM bound protects that customer from overloads from all other customers.

In a simple network, where there are only two customers, which might each be a class of customers, a GM bound for one customer would act as an UL bound for the other customer. In that setting, there are actually only two distinct control parameters, either the two GM bounds or the two UL bounds. However, for a more realistic example with more than two classes of customers, the UL and GM bounds provide more sophisticated controls.

These UL and GM bounds provide what can be called *significant separation with sharing*, which will be referred to as *partial sharing*. It is easy to see that partial sharing with UL/GM bounds can include as special cases both *complete partitioning (CP)* and *complete sharing (CS)*. The CP case is equivalent to having separate resources for each customer, while the CS case is equivalent to not having any sharing bounds.

Thus, partial sharing with UL/GM bounds necessarily can perform as well as either CS and CP. Indeed, our object is to show that with appropriate UL/GM bounds, partial sharing performs significantly better than both CS and CP in many scenarios.

In order for the UL/GM bounds to be effective, the appropriate UL/GM bounds need to be identified in any application. This problem is solved by the development of two appropriate algorithms:

- First, we developed an algorithm to compute the blocking probability of each customer for any specific assignment of UL/GM bounds. (See the section "Algorithm to Compute Blocking Probabilities" towards the end of this paper).
- Second, we developed a heuristic search algorithm to find the UL/GM bounds that efficiently meet specified blocking probability requirements. (See the section "Search Algorithm for UL/GM Bounds" at the end of this paper.)

The concept of UL/GM bounds itself is not new. Indeed UL/GM bounds were considered to analyze a node in a store-and-forward computer network 15 years ago by Kamoun and Kleinrock,[9] but the UL/GM bounds have not received much attention since then. Our main contribution is to show how UL/GM bounds can be easily enforced and analyzed. The UL bounds can easily be enforced if the resource provider keeps track of the number of requests from each customer in service. It is not so obvious how to efficiently enforce the GM bounds, but it can be done. Panel 3 shows how to efficiently enforce both bounds.

We will also compare trunk reservation to partial sharing with UL and GM bounds. In some scenarios, TR performs significantly better than the UL/GM bounds, so it is an important control to consider. In other scenarios, however, the UL/GM bounds perform significantly better. A major advantage of the UL/GM bounds over TR is the ease with which one can calculate the exact customer blocking probabilities for UL/GM policies, even when there are many classes of customers.

Specifying Grades of Service. We also propose a new way to specify customer grades of service. The grade of service is partly specified by stipulating the UL and GM bounds, but as indicated above, the actual performance is usually characterized in terms of request blocking probabilities.

Nominal Blocking Requirements. The customary procedure is to specify *nominal blocking requirements* on customer requests under normal conditions. Normal conditions imply that no customer is in overload—that is, no customer is submitting requests at a higher rate than was established when the customer subscribed to the service.

Conditional Blocking Requirements. The proposed new procedure is to specify, in addition to nominal blocking requirements, *conditional blocking requirements* on customer requests, which are blocking requirements conditional on the other customers being in some pattern of overload. For example, a conditional blocking requirement could be imposed if a condition exists whereby any *one* other customer is submitting requests at x percent above its nominal rate. For another example, there could be a conditional blocking requirement imposed because *all* other customers are in overload—that is, each other customer is submitting traffic at a rate of y percent above its nominal rate.

In our previous example, the two classes of customers—video conferencing and voice calls—were based on the service provided. However, a customer class also could be based on the *quality* of service. Thus, the service quality that the customer subscribes to and pays for could determine the degree of blocking that the customer can experience.

The use of conditional blocking requirements, as well as nominal blocking requirements, may make it necessary to provide protection for *each* customer against possible overloads caused by other customers. It is important to note that this feature is not provided by trunk reservation. For example, with two customers, trunk reservation protects only one of the two customers against overloads by the other.

Next, we consider numerical examples to show how the UL/GM bounds perform.

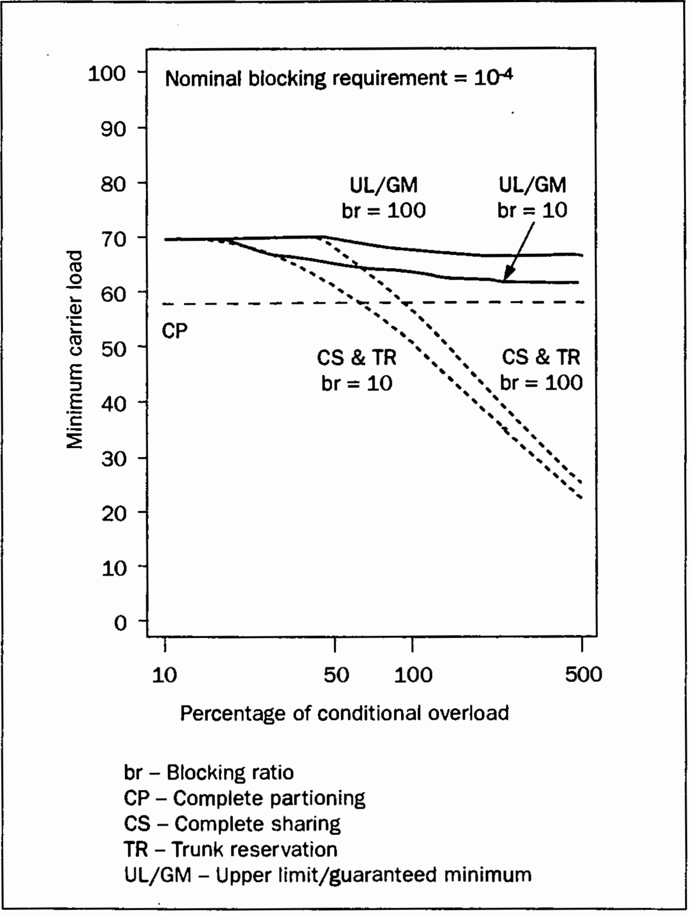

Figure 2. The maximum permissible arrival rate is shown as a function of the percentage of conditional overload for the CP, CS, and UL/GM sharing policies in the symmetric example. The nominal blocking requirement is 10^{-4} and the blocking ratio is equal to 10 or 100 for both customers.

Comparing Sharing Policies: Symmetric Examples

This section and the next present numerical examples to illustrate the value of partial sharing with the UL/GM bounds. These examples—made as simple as possible to focus attention on the sharing policies—have only a single resource and two customers (each of which could actually be a class of customers), whose requests require only a single resource unit. Since the case of two customers typically corresponds to two classes of customers—such as voice and dial-up video—here we refer to two customer classes.

Let's assume the resource has a capacity of 100 resource units and all the holding times have mean one time unit. (In actual applications, there is a wide range of possible time units, from milliseconds to seconds to hours—or even months.)

For those who wish to examine much larger examples that illustrate the power of the inversion algorithm, see two other papers.[10,11] There, the authors discuss a single resource with a capacity of 10^6 resource units, 3×10^4 customers, state-dependent service rates corresponding to finite multiserver queues, and customer requests that require multiple resource units. In contrast, the examples in this paper are sufficiently small that they can be solved for any sharing policy by solving the global balance equations of the Markov chain (exploiting sparsity).

The literature available on different sharing policies assumes only nominal blocking requirements or that the objective is to maximize the long-run average revenue in one specified operating regime (see Chapter 4 of Ross[8]). In contrast, we consider conditional blocking requirements, and show that the UL/GM bounds are especially useful for efficiently satisfying the conditional blocking requirements.

The comparisons in this section are for symmetric examples: The two classes of customers have common traffic parameters and common blocking requirements, and all requests require only a single resource unit.

The purpose of the experiment was to determine the maximum possible total arrival rate that satisfies the blocking requirements for each sharing policy. The two customer classes always contribute equally to this total rate—that is, the arrival rate of each is one half the total rate.

For symmetric examples, the optimal TR sharing policy always coincides with the CS sharing policy. This is intuitively reasonable, since trunk TR gives preference to one class over the other and, in a symmetric example, there is no reason to give such preference. Moreover, for symmetric models, the optimal CP policy is always an even split of the 100 resources.

With two customer classes, the two UL bounds are equivalent to the two GM bounds, so that there are only two relevant control parameters. Moreover, for symmetric models, the optimal GM parameters are equal. Hence, in this example, the UL/GM control is specified by a single GM number L, which is applied to each class. The number of resource units being shared is, thus, $100 - 2L$.

If there are only nominal blocking requirements—that is, if there are no conditional blocking requirements, then CS is obviously optimal. Suppose that we set the nominal blocking requirements at 10^{-4}, that is, 10^{-4} is the allowed blocking probability. Then we find that the CS policy supports a total rate of 69.3 arrivals per time unit, while the CP

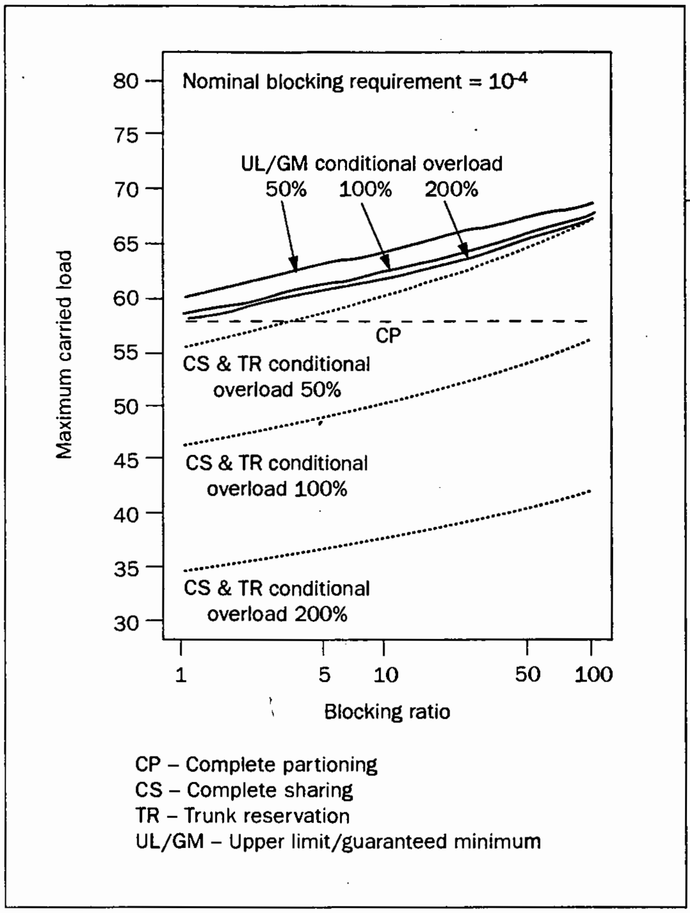

Figure 3. The maximum permissible arrival rate is shown as a function of the blocking ratio (br)—the ratio of conditional to nominal blocking requirement, the same for both customers—for the CP, CS, and UL/GM sharing policies in the symmetric example. The nominal blocking requirement is 10^{-4} and the conditional overload is equal to 50 percent, 100 percent, and 200 percent in the different cases.

policy supports a total arrival rate of 57.7. As indicated above, without conditional blocking requirements, the optimal UL/GM policy coincides with CS.

Now suppose that we introduce conditional blocking requirements. The question is: What arrival rates can the resource support between 57.7 for all conditional blocking requirements with CP, and 69.3 for CS without conditional blocking requirements?

In addition to the nominal blocking requirement of 10^{-4}, we now require a conditional blocking probability for each class when the other class is in overload. In particular, let's assume that the blocking probability for each class must be at most when the other class has an arrival rate 100 percent above its nominal arrival rate.

There are now two constraints, nominal (10^{-4}) and conditional (10^{-2}). As indicated above, we find that CP still supports a total arrival rate of 57.7, because the nominal constraint is binding, but CS now admits an arrival rate of only 56.0, which is a reduction of 19.2 percent from the previous CS value of 69.3.

On the other hand, the UL/GM policy with a GM bound *L* of 43 supports a total arrival rate of 67.2, which is only 3 percent below the CS value of 69.3 without the conditional blocking requirements. In this case, partial sharing with the UL/GM bounds permits a 16.5 percent higher arrival rate than CP and a 20.0 percent higher arrival rate than CS.

More generally, the advantage of UL/GM over CS and CP depends on two parameters: the blocking ratio (*br*) and conditional overload parameters (*x*). The *blocking ratio* is the ratio of the conditional blocking requirement to the nominal blocking requirement; normally *br* is greater than 1; here $br = 10^{-2}/10^{-4} = 100$. The *overload parameter* is the percentage overload allowed for the conditional blocking; here $x = 100$.

Assuming that the conditional blocking requirement is greater than the nominal blocking requirement, the CP policy is independent of the conditional blocking requirement. On the other hand, the total arrival rate supported by both CS and UL/GM decreases as the conditional overload increases and the blocking ratio decreases. Figure 2 shows how the CS and UL/GM policies depend on the conditional overload parameters for two values of the blocking ratio, 10 and 100. Figure 3 shows how they depend on the blocking ratio for three values of conditional overload, 50, 100, and 200. It should be noted that the optimal GM parameter *L* changes as well, also increasing in conditional overload and decreasing in the blocking ratio, but the change is not fast.

When the blocking ratio is very large or the conditional overload is very small, UL/GM behaves like CS. When the blocking ratio is very small or the conditional overload is very large, UL/GM behaves more like CP. However, as the conditional overload moves to infinity, UL/GM can perform much better than CP, because the GM bound alone can provide adequate protection against infinite overload from other customers. Indeed, the case of 500 percent conditional overload in Figure 2 closely approximates infinite conditional overload.

Figures 2 and 3 clearly show that there is a substantial region where UL/GM is significantly better than the best of CS and CP. Since CS coincides with the best TR policy for these symmetric examples, we see that

Table 2. Maximal Permissible Arrival Rate as a Function of the Sharing Policy and Overload Parameters.

Overload parameters		Resource sharing policy							
50 % conditional overload		CS	CP	TR	t_1	UL/GM	L_1	L_2	UL/GM (41,53)
Blocking ratio *br*	1	48.6	55.3	55.4	4	56.8	40	54	55.9
	3	50.0	55.3	57.5	4	58.5	41	54	57.5
	10	51.7	55.3	60.5	5	59.9	39	53	59.4
	30	53.3	55.3	64.5	5	61.3	38	52	59.7
	100	55.4	55.3	68.3	5	61.5	38	51	59.8
100 % conditional overload									
Blocking ratio *br*	1	40.5	55.3	46.1	4	55.9	43	55	53.8
	3	41.7	55.3	48.1	4	57.1	40	54	55.8
	10	43.1	55.3	50.8	4	58.1	41	53	58.1
	30	44.5	55.3	54.0	4	60.0	40	53	59.6
	100	46.2	55.3	58.9	4	61.4	38	51	59.6
200 % conditional overload									
Blocking ratio *br*	1	30.4	55.3	34.9	3	55.3	44	56	52.5
	3	31.3	55.3	36.4	3	56.9	43	55	54.7
	10	32.3	55.3	38.4	3	58.5	42	54	57.2
	30	33.4	55.3	40.7	3	60.0	40	53	59.7
	100	34.6	55.3	44.5	3	61.6	38	52	59.5

br — blocking ratio parameter
CP — complete partitioning
CS — complete sharing
TR — trunk reservation
UL/GM — upper limit/guaranteed minimum

UL/GM can be significantly better than TR as well.

Comparing Sharing Policies: Asymmetric Examples

This section considers the two-customer-class single-resource example introduced in the last section, but with different blocking requirements for the classes. In particular, the nominal blocking requirements are 10^{-3} for one class and 10^{-6} for the other. Various values are considered for the blocking ratio (again, the ratio of conditional-to-nominal blocking requirements), keeping this ratio the same for both customers.

The asymmetry now makes trunk reservation a more attractive alternative, so that our main objective is to compare TR to UL/GM, although CS and CP also are considered as before. The UL/GM policy is now specified by giving the two GM parameters L_1 and L_2. In our example, the number of resource units being shared is, thus, $100 - L_1 - L_2$.

Just as for UL/GM, a search algorithm is used to find the optimal TR parameter. For TR, there are two thresholds such that a request from each customer class can be admitted only if the number of free resource units after admission is at least the threshold for that class. Necessarily, the threshold should be zero for one class; otherwise, resource units would be wasted. When the threshold for both classes is zero, TR coincides with CS. Hence, the optimal TR policy always performs at least as well as CS.

For this simple example, in which the requirements and service rates of the two classes are identical, the TR blocking probability is easy to compute, because it is not necessary to keep track of the customer identity of requests in service. Hence, the model is what is called a birth-and-death process, which is readily solvable. For more general models, calculating the TR blocking probability can be much more difficult.

The experiment consists of 15 cases, with all

Table 3. The Proportions of the Four Blocking Requirements Realized for Different Policies

Sharing policy	Proportion of blocking requirement realized			
	Nominal		Conditional	
	100 % conditional overload, br = 1			
CS	0.13×10^{7}	0.13×10^{4}	0.98×10^{3}	0.98
CP	1.00	0.74	1.00	0.74
TR	0.13×10^{6}	0.70	1.00	0.38
UL/GM	0.73	0.49	0.87	0.95
	100 % conditional overload, br = 10			
CS	0.41×10^{7}	0.41×10^{4}	1.00×10^{3}	1.00
CP	1.00	0.74	0.10	0.074
TR	0.013	0.11×10^{4}	1.00	0.57
UL/GM	0.55	0.42	0.20	0.84
	200 % conditional overload, br = 100			
CS	0.25×10^{7}	0.25×10^{4}	1.00×10^{3}	1.00
CP	1.00	0.74	0.10	0.074
TR	0.10×10^{7}	0.20×10^{7}	0.98	1.00
UL/GM	0.96	0.89	0.22	0.93
	50 % conditional overload, br = 100			
CS	0.20×10^{4}	0.020	0.97×10^{3}	0.97
CP	1.00	0.74	0.01	0.0074
TR	0.84	1.00	0.65	0.58
UL/GM	0.58	0.93	0.015	0.39

br — blocking ratio parameter
CP — complete partitioning
CS — complete sharing
TR — trunk reservation
UL/GM — upper limit/guaranteed minimum

combinations of three values of the conditional overload and five values of the blocking ratio: the percentage of conditional overload is 50 percent, 100 percent, or 200 percent, and the blocking ratio (the same for both customers), is 1, 3, 10, 30, or 100.

The maximal total arrival rates as a function of the sharing policy chosen for these 15 cases are given in Table 2. The optimal control parameters t_1 for TR and L_1 and L_2 for UL/GM are also given in Table 2. Also displayed in Table 2 are the maximal arrival rates for a fixed UL/GM sharing policy with GM parameters $L_1 = 41$ and $L_2 = 53$. These are the optimal GM parameters for the "middle" case of 100 percent conditional overload and a conditional-to-nominal ratio of 10. This fixed UL/GM sharing policy demonstrates the robustness of the UL/GM bounds. Even though this sharing policy is not optimal in the other 14 cases, it performs pretty well overall.

In this asymmetric case, the best CP policy is no longer an even split. Now for the best CP policy, the capacities of the two separate resources are 44 and 56. As in the previous section, the performance of the CP policy is the same for all 15 cases. In the case of a 200 percent conditional overload and a blocking ratio of 1, the best UL/GM policy coincides with the CP policy. But in all other cases, the best UL/GM policy performs better than the CP policy.

Necessarily, in all these other cases the UL/GM policy allows for some sharing. In the 15 cases, the number of resource units shared with the UL/GM bounds ranges from 0 to 10. The maximum advantage of UL/GM over CP is 11.4 percent, occurring in the case of most sharing, with a 200 percent conditional overload and a blocking ratio of 100 for both customers.

With one exception, where the difference is negligible, CP dominates CS in all these cases. Thus, TR also dominates CS. However, TR can be much inferior to CP, and thus also to UL/GM. In the case of a 200 percent conditional overload and a blocking ratio of 1, the best UL/GM policy produces a 58 percent higher arrival rate than the best TR policy. However, in the case of only a 50 percent conditional overload and a blocking ratio of 100, the best TR policy produces an 11 percent higher arrival rate than the best UL/GM policy. Evidently both UL/GM policies and TR policies, and possibly other policies not considered here, are useful controls in this context.

Table 3 displays the proportions of the blocking requirements realized by the different policies. The better policies realize proportions relatively close to one for all four requirements listed in the table. The policies that perform poorly tend to have some very small proportions.

In summary, the examples demonstrate the value of the UL/GM bounds. In many cases, partial sharing with the UL/GM bounds significantly outperforms CS, CP, and TR.

Applications of the Algorithms

The algorithms for calculating customer request blocking probabilities with UL/GM bounds, and finding appropriate UL/GM bounds, make it possible to address several important engineering problems.

Real-Time Customer Admission Control. A successful scheme for efficiently providing multiple grades of service should address the problem of *real-time control* of customer admission. For given limited resources, the resource provider needs to be able to determine whether or not each prospective new customer can be admitted. The algorithm can be used to quickly determine whether the new customer can be given the desired grade of service, while ensuring all previously admitted customers still receive their grades of service.

Capacity Adjustment. Over time, the level of customer demand often changes, making it necessary to adjust the resource capacity. The proposed algorithm can be used to determine this required capacity.

Demand for service may grow or decline, or there also may be a temporary reallocation of demand due to a resource failure. In some such resource failure situations, customers can be assigned to alternative resources, although this may increase demand on those resources to the point where the resource provider has to add capacity to these alternative resources.

However, in some emergencies, additional capacity may not be provided quickly enough, requiring the resource provider to provide the best possible service in the face of a reduced resource capability. If customers who were using a failed resource can be assigned to alternative resources, then ways are needed to protect the original customers on these alternative resources from the newly diverted demand, while protecting the diverted customers as well.

There are many methods to identify alternative resources available to serve customers that had been using a failed resource. In some situations, the alternative resources may be evident. For example, the system may contain only two resources, with each serving as a backup for the other. In other settings, automatic procedures could be used to dynamically reallocate demand to new resources, as is the case with schemes for alternative routing of blocked calls, as in AT&T's telephone network. Another possibility is using a special procedure invoked by a central controller. Where there is no centralized control, a distributed algorithm may be needed to first inform all resources that a failure has taken place, and then to set up appropriate alternative routes.

Regardless of the method used to generate alternative resource assignments, there is a need to provide some protection for both the original and diverted customers on the remaining resources. Here the use of UL/GM bounds, perhaps applied to classes of customers, would be a solution. The algorithm can be used to help determine the UL/GM bounds and the amount of traffic to divert.

The Costs of Providing Service. When considering potential schemes for providing multiple grades of service with protection against overloads, the resource provider may want to assess the costs of providing given grades of service, so that an effective pricing scheme can be developed. There are several possible ways to determine the cost of providing service to a new customer, even if one focuses only on the capacity used on each resource. Clearly, the minimum capacity used is the GM bound, and the maximum capacity used is the UL bound. Thus, one should look for a notion of *expected capacity used* by a customer, which necessarily will fall between these two extremes.

Marginal Expected Cost. One expected cost expression is the *marginal expected cost*, which is the cost for the extra capacity required beyond what is required for all other customers who submit requests according to their specified parameters. The marginal expected cost can be determined by first finding the minimum capacity of the resource required to meet all current customer requirements, and then finding the minimum capacity of the resource to meet all current customers' requirements *plus* those of the new customer. The marginal expected cost for the new customer is the difference between these two capacity levels. The algorithm can be used to determine the two critical capacity levels, just as in the capacity adjustment procedure.

First-Customer Expected Cost. Another expected cost is the *first-customer expected cost*, which is the average capacity used if that customer were the only customer using the resource. This cost can be determined by finding the minimum capacity of the resource needed to meet the customer's requirements, assuming that no other customers are present and that this customer submits requests according to its agreed-upon traffic parameters.

The first-customer expected cost will typically be higher than the marginal expected cost, because there is less sharing. The first-customer expected cost also can be determined by invoking the algorithm, just as with the marginal expected cost. The calculation, of course, is easier with only a single customer.

Algorithm to Compute Blocking Probabilities

Now let us discuss—at a high level—the numerical transform inversion algorithm used to compute the blocking probabilities. The basic theory for resource-sharing models with the UL/GM bounds implies that a steady-state distribution of the number of requests in service for each customer in the model has a tractable product-form.[1,2,8] The basic theory is reviewed in Panel 2. An important point is that the UL/GM bounds constitute a special case of a *coordinate-convex sharing policy*, which has a product-form steady-state distribution just like the complete-sharing policy.[1,2,8] (TR is more difficult to analyze because it is not a coordinate-convex policy and does not have such a product-form steady-state distribution.)

The product-form property for partial sharing with the UL/GM bounds implies that the steady-state distribution is readily available except for a normalization constant (or partition function). Moreover, the desired customer-request blocking probabilities can be expressed simply as ratios of two normalization constants. Unfortunately, however, because the models get large, the normalization constants themselves quickly become difficult to compute. Various approaches have been proposed for resource-sharing models, including recursive algorithms,[2-4,8,11] Monte-Carlo algorithms,[8] and approximations,[5,7,8] but none of these have addressed models with UL and GM bounds.

Numerical Transform Inversion. We propose a new approach: *numerical transform inversion*.[10,11] It turns out that it is possible to construct the generating function (or z-transform) of the normalization constants in a remarkably compact form. We calculate the normalization constants by numerically inverting the generating function using our recently developed variant of the Fourier-series method,[12,13] along with special scaling for error control.[10,11]

Unfortunately, however, the numerical inversion algorithm can also have high computational complexity, tending to be exponential in the dimension of the generating function. Hence, the current upper limit on the dimension amenable for computation is about five. It thus might appear that there is little hope for the inversion algorithm, because with UL and GM bounds for each customer, the dimension of the generating function is $p + 2r$, where p is the number of resources and r is the number of customers. (See Panel 2)

Dimension Reduction. The key to successful inversion with large models is reducing the effective dimension. Because of the way the UL/GM bounds affect the generating function structure, the effective dimension can always be reduced from $p + 2r$ to $p + 2$, $p + 1$, or p by inverting the variables in a good order.[10,11] Thus, it is always possible to solve a resource-sharing model with only a few resources, even if there are hundreds or thousands of customers.

Moreover, with more resources, it is often possible to further reduce the effective dimension by:

- Doing an initial approximate analysis to eliminate very lightly loaded resources,[11] and
- Taking advantage of partial independence in the generating function structure, just as for the UL/GM bounds.[10]

Other Measures. Even after the effective dimension has been reduced to a manageable size, there are

Panel 2. The Mathematical Resource-Sharing Model

We briefly describe the mathematical resource-sharing model, including the product-form steady-state distribution, the normalization constants, and the blocking probability expressions. For more details, see the references.[10,11] Let there be *p resources* and *r customers*. Let the resources be indexed by i and the customers by j. Let resource i have K_i units, $1 \le i \le p$, and let $\mathbf{K} \equiv (K_1,..., K_p)$ be the *capacity vector*. (Let vectors be row vectors or column vectors; it will be clear from the context.) Each customer *j request* requires a_{ij} units on resource i, where a_{ij} is a deterministic nonnegative integer. Let $\mathbf{A}$ be the *p x r requirements matrix* with elements a_{ij}. Let the system state vector be $\mathbf{n} \equiv (n_1,..., n_r)$, where n_j is the number of customer-j requests in service.

In order to treat the GM bounds, we assume that a_{ij} is either b_j or 0 for all i. Let $\delta_{ij} = 1$ if $a_{ij} > 0$, and let $\delta_{ij} = 0$ otherwise. Let L_j be the ***guaranteed-minimum*** bound (L for lower bound) on the number of requests for customer j. Let $N_j = b_j L_j$ and let $\mathbf{N} = (N_1,..., N_r)$. Let U_j be the upper limit for customer-j requests and let $\mathbf{U} \equiv (U_1,..., U_r)$.

The state space, the space of allowable states, is a subset of $\mathbf{Z}^r_+$, the r-fold product of the nonnegative integers. Let $S(\mathbf{K}, \mathbf{U}, \mathbf{N})$ be the state space, reflecting its dependence upon the vector $(\mathbf{K}, \mathbf{U}, \mathbf{N})$. It is

$$S(\mathbf{K}, \mathbf{U}, \mathbf{N}) = \{\mathbf{n} \in \mathbf{Z}^r_+ : \mathbf{n} \le \mathbf{U}, \sum_{j=1}^{r} (a_{ij} n_j \vee \delta_{ij} N_j) \le K_i, 1 \le i \le p\}$$

where $x \vee y = \max\{x, y\}$. The first set of constraints $\mathbf{n} \le \mathrm{U}$ represents the UL bounds; and the second set of constraints,

$$\sum_{j=1}^{r} (a_{ij} n_j \vee \delta_{ij} N_j) \le K_i, 1 \le i \le p,$$

represents the GM bounds. Note that the GM constraints also include the basic CS constraints $\mathbf{An} \le \mathbf{K}$. Without the UL/GM bounds, the state space reduces to $S(\mathbf{K})$ with only the constraints $\mathbf{An} \le \mathbf{K}$.

We now specify the stochastic arrival and service processes. We assume that the stochastic process $\{\mathbf{N}(t) : t \ge 0\}$, where $\mathbf{N}(t)$ gives the system state at time t, is an irreducible finite-state continuous-time Markov chain. This Markov chain is specified by giving the arrival and service rates for all customer requests. We allow general state-dependent arrival and service rates. In particular, if there are k customer-j requests in service, then the ***arrival rate*** of customer-j requests is $\lambda_j(k)$ and the ***service-completion*** rate of customer-j requests is $\mu_j(k)$. The standard case is $\lambda_j(k) = \lambda_j$ and $\mu_j(k) = k\mu_j$, which corresponds to Poisson arrivals and exponential service times with all admitted requests entering service immediately upon arrival. Our numerical examples are all for this special case. State-dependent arrival and service rates greatly increase the power of the model; for example, state-dependent arrival rates can be used to approximate non-Poisson overflow traffic arising in alternative routing schemes.[11]

We assume that each request is admitted if all desired resource units can be provided, without violating any constraints; otherwise the request is blocked and lost. We do not consider retrials. All resource units used by a customer request are released at the end of the request holding time.

(Continued on next page.)

additional steps that are taken to reduce the computational complexity of the inversion algorithm.[10,11] The algorithm is made more efficient by:

- Selective truncation of large sums without loss of accuracy when the resource capacities are large,
- Efficiently treating multiple customers with identical parameters, and
- Sharing computation of normalization constants when there are many customers and large capacities.

Thus, many customers and large capacities do not pose great problems for the algorithm.

If we have a large, highly connected network of fully utilized resources, then it may not be possible to reduce the effective dimension sufficiently for the inversion to apply directly. In that case, we can apply *reduced-load fixed-point approximations* to calculate the blocking probabilities approximately.[5,8] With the fixed-point approximation, the inversion algorithm can be applied as a subroutine to solve the single-resource models with UL/GM bounds exactly. Moreover, the inversion algorithm makes it possible to base the fixed-point approximation on larger subnetworks, each containing two or three resources.

Search Algorithm for UL/GM Bounds

The inversion algorithm just described makes it possible to calculate request blocking probabilities, given customer traffic parameters and UL/GM bounds. To address important engineering applications, however, the appropriate UL/GM bounds must first be determined. For this purpose, a heuristic search algorithm is used. Its effectiveness depends on being able to solve relatively quickly the request blocking probabilities for any given set of traffic parameters and UL/GM bounds. With the aid of the inversion algorithm, one can locate the most effective UL/GM parameters by solving, via fine tuning, many instances of the model.

For example, suppose that we wish to determine

The *steady-state probability vector* π has the simple product form

$$\pi(\mathbf{n}) = g(\mathbf{K},\mathbf{U},\mathbf{N})^{-1} f(\mathbf{n}),$$

where

$$f(\mathbf{n}) = \Pi_{j=1}^{r} f_j(n_j),$$

$$f_j(n_j) = \Lambda_j(n_j) / M_j(n_j),$$

$$\Lambda_j(n_j) = \Pi_{k=0}^{n_j-1} \lambda_j(k),$$

$$M_j(n_j) = \Pi_{k=1}^{n_j} \mu_j(k),$$

and the *normalization constant* or *partition function* is

$$g(\mathbf{K},\mathbf{U},\mathbf{N}) = \sum_{\mathbf{n} \in S(\mathbf{K},\mathbf{U},\mathbf{N})} f(\mathbf{n})$$

The term *product form* refers primarily to the product form of $f(\mathbf{n})$.

It is significant that the request blocking probabilities can easily be expressed in terms of the normalization constants. However, it is important to distinguish between *request blocking* and *time blocking*. Request blocking refers to the blocking experienced by arrivals, which depends on the state at arrival epochs, while time blocking refers to the blocking that would take place at an arbitrary time if there were an arrival at that time.

Since the steady-state distribution π refers to an arbitrary time, blocking probabilities computed directly from it involve time blocking, but it is not difficult to treat request blocking, as well as time blocking. With Poisson arrivals, the two probability distributions at arrival epochs and at an arbitrary time agree, but not more generally.[14] The probability that a class-j request would not be admitted at an arbitrary time (time blocking) is

$$B_j^t = 1 - \frac{g(\mathbf{K}-\mathbf{A}\mathbf{e}_j, \mathbf{U}-\mathbf{e}_j, \mathbf{N}-\mathbf{A}\mathbf{e}_j)}{g(\mathbf{K},\mathbf{U},\mathbf{N})}$$

where $\mathbf{e}_j$ is a vector with a 1 in the jth place and 0's elsewhere. It turns out that the customer-j request blocking B_j is just the time blocking B_j^t for the modified model in which the customer-j arrival-rate function is changed from

$$\lambda_j(m) \text{ to } \bar{\lambda}_j(m) = \lambda_j(m+1).^{10}$$

Hence, in order to calculate the desired blocking probabilities, it suffices to calculate the normalization constants $g(\mathbf{K},\mathbf{U},\mathbf{N})$ for various vector arguments $(\mathbf{K},\mathbf{U},\mathbf{N})$. For this purpose, we construct the generating function of the normalization constant as a function of $(\mathbf{K},\mathbf{U},\mathbf{N})$ and invert it.[10,11]

how much demand can be satisfied by resources with given capacities. This problem can be addressed by trying to maximize the customers' request arrival rates—subject to all the blocking requirements—assuming that the proportions of the total arrival rates from the different customers are known. The first step is to use a binary search to find the maximum feasible total arrival rate with a complete-sharing policy, that is, with no bounds. Then, a local search algorithm is used to look for the appropriate UL/GM bounds for each customer class that would allow a further increase of the arrival rate by a suitably small incremental change, usually by 1 percent. The total arrival rate can continually be increased incrementally in this way until no further improvement is possible by modifying the UL/GM bounds.

Similarly, for a single resource, we may be interested in minimizing the capacity of that resource, subject to a given set of users with specified traffic parameters and blocking requirements. As discussed above, we start by using a binary search to find the minimum feasible capacity with the CS policy. We then use a local search to find UL/GM bounds that allow us to reduce the capacity further, until no further reduction in capacity is possible.

Conclusion

We have shown that the UL and GM bounds make it possible to efficiently provide multiple grades of service with protection against overloads in shared resources. We have focused on shared resources arising in multimedia telecommunications systems, but there clearly are many other possible applications as well.

We have compared the UL/GM bounds to the traditional controls of CS, CP, and TR. Numerical examples show that UL/GM can significantly outperform these other alternatives, although in some scenarios TR is a better control.

The key to the effectiveness of the UL/GM bounds is the ability for them to be efficiently enforced and analyzed. Efficient analysis is made possible by a new algorithm for computing blocking probabilities based on numerically inverting the generating function of the normalization constant.

Panel 3. Enforcing the Bounds

Providing multiple grades of service with the UL and GM bounds requires that these bounds be enforced on an ongoing basis. We describe a procedure for efficiently enforcing these bounds using the notation of Panel 2.

Key variables are the number n_j of active requests for customer j and the number F_i of free units in resource i. The initial numbers for an empty system are $n_j = 0$ and

$$F_i = K_i - \sum_{j=1}^{r} L_j a_{ij} .$$

When a new customer-j request arrives, the new request is admitted if *both* $n_j \leq U_j - 1$ and, for each i such that $\mathrm{a_{ij}} > 0$,

$$(n_j + 1)\mathrm{a_{ij}} \leq F_i + L_j a_{ij}.$$

If these conditions are not satisfied, then the request is not admitted. If the customer-j request is admitted, then the variables are updated by first increasing n_j by 1 and then, for each i such that $a_{ij} > 0$, decreasing F_i by a_{ij} if $n_j > L_j$, and making no change otherwise.

When a customer-j request service completion occurs, these variables are updated by first decreasing n_j by 1 and then, for all i such that $a_{ij} > 0$, increasing F_i by a_{ij} if $n_j \geq L_j$, and leaving it unchanged otherwise.

When the system starts with some initial numbers of requests in service, the initial number of free resource units must be computed for each i using

$$F_i = K_i - \sum_{i=1}^{r} \max\{L_j a_{ij}, n_j a_{ij}\}.$$

The overall computational complexity of the above expression is $O(rp)$. However, at each subsequent customer-j request arrival-or-departure epoch, the computational complexity is only $O(q_j)$, where q_j is the number of resources used by customer j. In applications r and p may both be large, while q_j is small for all j. Accordingly, there may be the one-time large substantial start-up computation, but at each subsequent request arrival-or-departure epoch the amount of computation is low, being of the same order as for other simple request admission policies such as complete sharing.

(Manuscript approved May 1995)

References

1. F. P. Kelly, *Reversibility and Stochastic Networks*, Wiley, New York, 1979.
2. J. S. Kaufman, "Blocking in a Shared Resource Environment," *IEEE Transactions on Communications*, Vol. 29, 1981, pp. 1474-1481.
3. Z. Dziong and J. W. Roberts, "Congestion Probabilities in a Circuit-Switching Integrated Services Network," *Performance Evaluation*, Vol. 7, 1987, pp. 267-284.
4. D. Tsang and K. W. Ross, "Algorithms to Determine Exact Blocking Probabilities for Multirate Tree Networks," *IEEE Transactions on Communications*, Vol. 38, 1990, pp. 1266-1271.
5. F. P. Kelly, "Loss Networks," *Annals of Applied Probability*, Vol. 1, 1991, pp. 319-378.
6. H. A. B. van de Vlag and G. A. Awater, "Exact Computation of Time and Call Blocking Probabilities in Multi-Traffic Circuit-Switched Networks," *Proceedings of IEEE Infocom '94*, 1994, pp. 56-65.
7. D. Mitra and J. A. Morrison, "Erlang Capacity and Uniform Approximations for Shared Unbuffered Resources," *IEEE/ACM Transactions on Networking*, Vol. 2, 1994, pp. 558-570.
8. K. W. Ross, *Multiservice Loss Models for Broadband Telecommunication Networks*, 1995, to appear.
9. F. Kamoun and L. Kleinrock, "Analysis of Shared Finite Storage in a Computer Network Node Environment Under General Traffic Conditions," *IEEE Transactions on Communications*, Vol. 28, 1980, pp. 992-1003.
10. G. L. Choudhury, K. K. Leung, and W. Whitt, "An Algorithm to Compute Blocking Probabilities in Multi-Rate Multi-Class Multi-Resource Loss Models," *Advances in Applied Probability*, 1995, to appear in December. (Abbreviated version in *Proceedings of IEEE Globecom '94*, pp. 1123-1128.)
11. G. L. Choudhury, K. K. Leung, and W. Whitt, "An Inversion Algorithm to Compute Blocking Probabilities in Loss Networks with State-Dependent Rates," *IEEE/ACM Transactions on Networking*, to appear. (Abbreviated version in *Proceedings of IEEE Infocom '95*, Boston, 1995, pp. 513-521.)
12. J. Abate and W. Whitt, "The Fourier-Series Method for Inverting Transforms of Probability Distributions," *Queueing Systems*, Vol. 10, 1992, pp. 5-88.
13. G. L. Choudhury, D. M. Lucantoni, and W. Whitt, "Multidimensional Transform Inversion With Applications to the Transient M/G/1 Queue," *Annals of Applied Probability*, Vol. 4, 1994, pp. 719-740.
14. B. Melamed and W. Whitt, "On Arrivals That See Time Averages," *Operations Research*, Vol. 38, 1990, pp. 156-172.

Gagan L. Choudhury is a distinguished member of technical staff in the Teletraffic Theory and System Performance Department of AT&T Bell Laboratories in Holmdel, New Jersey. He is responsible for performance analysis of telecommunications systems, and recently has focused on numerical transform inversion. He joined the company in 1982. He has a B.S. degree in radio physics and electronics from the University of Calcutta, India, and an M.S.E.E. degree and a Ph. D. in electrical engineering, both from the State University of New York, Stony Brook.

Kin K. Leung is a distinguished member of technical staff in the Teletraffic Theory and System Performance Department of AT&T Bell Laboratories in Holmdel, New Jersey. He is involved in performance analysis and architectural design of wireless networks, communications networks, and computer systems. He joined the company in 1986. He has a B.S.E.E. degree from the Chinese University of Hong Kong, and an M.S. degree and a Ph. D., both in computer science, from the University of California, Los Angeles.

Ward Whitt is a distinguished member of technical staff in the Network Services Research Center of AT&T Bell Laboratories in Murray Hill, New Jersey. His interests include probability theory, queueing models, numerical transform inversion, and performance analysis of networks. He joined the company in 1977. He has an A.B. degree in mathematics from Dartmouth College in Hanover, New Hampshire, and a Ph.D. in operations research from Cornell University in Ithaca, New York.