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

by

Gagan L. Choudhury, Kin K. Leung and Ward Whitt

Field of the Invention

The invention relates to efficiently providing multiple grades of service, including protection against overloads, for multiple customers sharing limited resources, wherein "efficiently providing" includes (i) determining when a prospective new customer can be admitted with a desired grade of service, (ii) determining how to adjust capacity in face of changing customer demand and (iii) determining how to respond to resource failure.

Background of the Invention

1. Resource-Sharing Problems

The setting involves one or more *resources*, each containing multiple resource *units* which provide service to multiple *customers*. Each customer is a source of a stream of *requests*. Each customer request requires a number of units from each resource, which may be zero, one or greater than one, and may be different for different customers, but which is the same for different requests of the same customer. If all requirements can be met upon arrival of a new request, then the new request is admitted, and all required resource units are held throughout the request holding time. Otherwise, the request is not admitted, and is said to be *blocked*.

Resource sharing as described above takes place in many physical settings, for example, in communication networks and in computer systems; for background see the references, wherein resource-sharing models are often called "loss networks". In a circuit-switched telecommunications network, the resources may be *links* and the resource units may be *circuits* on these links, while the customers may be associated with different *services* (e.g., voice, data, video and facsimile), and the requests may be *calls*. Calls often require circuits simultaneously on several different links, with the specific links depending on the origin and destination of the call. For some services, calls require multiple circuits on each link.

A circuit-switched telecommunication network example is depicted Figure 1. In this figure there are 5 nodes and 5 links, with link i having K_i circuits. (In this case the nodes play no role.) 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 setting of Figure 1 there might be six routes requiring the following subsets of links: $\{1\}$, $\{2\}$, $\{1,2\}$, $\{3,5\}$, $\{4,5\}$ and $\{1,3,5\}$. There might be many more customers, because there can be different customers on the same route. For example, some customers may require only one circuit on each link, whereas other customers may require six circuits on each link.

In a broadband integrated-services digital network (B-ISDN) supported by the asynchronous transfer mode (ATM) technology, the resources may be *switches* and other network facilities, the resource capacity (units) may be the *bandwidth* available at these network facilities, while the customers may be prospective *users* of the network who wish to establish a connection and the customer requests may be ATM cells, bursts of ATM cells or required "effective bandwidths" associated with bursts of ATM cells within an established connection. Figure 1 could also apply to this B-ISDN example. If the critical resources are the switches, then the nodes are the resources. More generally, both nodes and links might be important resources. Since B-ISDN networks are intended for a broad range of services, it is important to allow customers to have very different traffic characteristics and very different requirements. In particular, requests from different customers may use different numbers of resource units. In this context, admission control is a well known challenging problem of current interest; see Roberts [9].

2. The Probabilistic Nature of the Problem

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 no more capacity than necessary. The second is the probabilistic nature of the problem. The submission of customer requests and the holding times of these requests are uncertain events that fluctuate over time. Therefore, 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 part of the current state of the art to use probabilistic or stochastic models to characterize the customer requirements. In this uncertain environment with limited resources, some blocking of customer requests becomes inevitable unless the resource capacity greatly exceeds demand. It is thus customary to characterize the quality of service received in terms of a customer's *request blocking probability*, i.e., the long-run proportion of requests that are blocked in a particular operating regime. Each customer wishes to have its request blocking probability be suitably low. Request blocking probabilities that are two high fail to satisfy customer requirements. On the other hand, request blocking probabilities that are too low mean that more capacity has been provided than is needed.

In this probabilistic setting, the resource provider needs to: (i) determine the admissibility of a given set of customers for given resource capacities, (ii) determine appropriate resource capacities for a given set of customers, and (iii) determine how to respond to temporary unavailability of one or more resources because of resource failure. These problems can be considered when each customer's request traffic is characterized (probabilistically) and each customer's blocking probability requirements are given. When all customers have full access to the resources, i.e., when a *complete-sharing policy* is used, there are methods available within the current state of the art to calculate the blocking probabilities for each class in a model of the system under consideration. The resource provider can then check to see if the prevailing customer-request blocking probabilities meet the requirements. The resource-provider can use these calculations to determine whether or not to admit new customers, to determine appropriate capacities, and to determine how to respond to resource failures. However, existing methods for computing these blocking probabilities encounter severe difficulties as the model gets large. Thus, there has remained a need to be able to solve much larger models of the standard form than could be solved up to now.

3. The Need for Protection Against Overloads

Moreover, the resource provider actually is strongly motivated *not* to use the complete-sharing policy, for which state-of-art methods often apply. An important feature of many emerging resources is the presence of different kinds of customers. In this setting, it is very important to protect the customers from each other. If all customers are allowed full access to the resource, then one or more customers may actually submit requests at very high rates, much above their negotiated rates, which can cause other classes to experience unacceptably high blocking probabilities.

Thus, some way is needed to *protect customers from overloads by other customers.* One way to do this is to *partition* the resources into separate portions dedicated to each customer, but this tends to be inefficient, because the benefits of sharing are lost. The overall capacity required in each resource tends to become unacceptably large with complete partitioning.

Another possible control scheme that allows some sharing is the rate-based multiple-class access-control scheme of Berger, Milito and Whitt (previous patent). That scheme regulates the requests admitted from different customers, but it is based solely on the pattern of requests. It does not take into account the holding times of these requests. More generally, it does not use the current state of the resources. In particular, that rate-based control does not depend on the number of requests from each customer currently in service. Here we consider situations in which this extra information is available to the resource provider. It is natural to look for controls which exploit this additional information when it is available.

Moreover, many schemes, such as the rate-based multiple-class access control scheme of Berger, Milito and Whitt, only regulate the flow of customer requests. The resource provider also needs to know whether or not to allow customers to send requests. The customer admission problem is also of major concern.

Instead of complete sharing or complete partitioning, it is natural to consider various *constraints* on the allowable states in the resources that still allow some sharing. One such scheme that is part of the current state of the art is *trunk reservation.* With two classes, trunk reservation has one of two classes be blocked whenever the total free capacity falls below a specified threshold. This scheme protects one of the two classes against overloads from the other, but it does not protect both classes. More generally, with n classes, a natural generalization of trunk reservation has the first k classes blocked if the total free capacity falls below a specified k^{th} threshold, where there are $n-1$ threshold values. However, just as in the two-class case, all classes are not protected with this scheme.

Moreover, with the current state of the art, it is relatively difficult to calculate the customer blocking probabilities with the trunk-reservation policy. The trunk-reservation policy causes the standard model to lose the nice product-form structure which makes it possible to analyze the complete-sharing and complete-partitioning policies. Other constraints also tend to cause severe problems in computing the blocking probabilities within the current state of the art. It is known that a large class of constraints, leading to the so-called "coordinate-convex" sharing policies, do possess product-form steady-state distributions; e.g., see Kaufman [7], but good algorithms for general coordinate convex policies have not been developed. Algorithms for computing customer request blocking probabilities with constraints have previously been developed in only a few very special cases.

4. The Need for Different Grades of Service

With different customers, it is not only important to protect against overloads of others, but it is also important to be able to provide *different grades of service.* Some customers may be willing to pay more to have lower blocking probabilities and stronger protection against overloads from others, while other customers would prefer to have higher blocking probabilities and weaker protection against overloads from others at a lower price.

Even achieving similar blocking probabilities is difficult, because different customers may require different numbers of resource units for each request. A customer whose requests require a larger number of resource units then another tends to have a higher blocking probability. Different grades of service can be achieved by complete partitioning, but what is needed are *more efficient* ways to provide different grades of service that allow sharing.

5. Associated Problems in Managing the Shared Resources

When considering potential schemes for providing multiple grades of service with protection against overloads, other associated problems arise. First, the resource provider wants to *assess the costs of providing given grades of service*, so that an effective *pricing scheme* can be developed. There are different ways to represent such costs. One important way is to determine the average capacity on each resource used by each customer. With any sharing policy that is used to provide different grades of service, the service provider would thus like to be able to calculate the average capacity used by each customer.

A successful scheme for efficiently providing multiple grades of service should address the problem of *real-time customer admission control.* For given limited resources, the resource provider needs to be able to determine whether or not each prospective new customer can be admitted. With multiple grades of service, the resource provider needs to determine whether the new customer can be given his desired grade of service, with all previously admitted customers still receiving their previously determined grades of service.

Any way of providing different grades of service presumably imposes bounds on the requests submitted by admitted customers. An efficient scheme for *enforcing these bounds* is needed. Since checking has to be done for each customer request, computational efficiency is of serious concern.

Over time, the level of customer demand often changes. There may be growing or declining demand for service. There may also be a temporary reallocation of demand in the face of *resource failure.* In some resource failure situations, customers can be served by assigning them to alternative resources, but this increases the customer demand on these alternative resources. In response, the resource provider may have the opportunity to add capacity to these other resources. In face of such changing customer demand, the resource provider needs a way to determine the amount of capacity needed to meet this demand. This is the *capacity adjustment problem.*

The resource provider often will have a more complicated set of options to respond to resource failure. As indicated above, customers might be able to receive service from an alternative set of resources. In some cases, this will require additional capacity that can be immediately provided at other resources, possibly at extra expense. However, in other cases, additional capacity cannot be provided quickly enough. Then the resource provider needs a way to provide the best possible service in face of the uncontrollable resource failure. 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 other resources from the diverted demand. At the same time, the resource provider would like to protect the diverted customers as well.

Within the state of the art, there are many methods to identify alternative available resources to serve customers that were using a failed resource. In some settings 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, there may be an automatic procedure in place to dynamically reallocate demand to new resources, as is the case with schemes for alternative routing of blocked calls in telecommunications networks. Another possibility is that a special procedure may need to be invoked by a central controller in the event of resource failure, as in the algorithm for finding alternate routing arrangements developed by Mansour and Nguyen (patent no. 5,058,105). 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 (resource assignments).

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.

6. Summary of the Problems

In summary, the resource provider is faced with these problems:

1. How to efficiently provide multiple grades of service with protection against overloads?
2. How to determine whether the grades of service can be provided to a given set of customers by resources with given capacities?
3. How to assess the cost of providing a given grade of service?
4. How to determine in real time whether or not each prospective customer with a desired grade of service can be (or should be) admitted?
5. How to determine the new capacities of the resources needed to satisfy a new set of customer demands?
6. How to respond to a resource failure?

Summary of the Invention

1. New Grades of Service

The invention is a new scheme for providing different grades of service with protection against overloads. This scheme is based on assigning each customer *upper-limit* (UL) and *guaranteed-minimum* (GM) *bounds* on its requests. The upper limit bound puts an upper limit on the number of requests from that customer that can be in service at any time. The guaranteed-minimum bound guarantees that there is always space for a specified number of requests from that customer. As special cases, this sharing policy includes complete sharing and complete partitioning, but it also includes significant separation with sharing. Indeed, it can be demonstrated that the UL/GM sharing policy outperforms complete partitioning and complete sharing.

Variations on the UL and GM bounds are also allowed. Alternative bounds can be based on other linear constraints on the number of active requests of different customers. These alternative bounds include UL bounds for *subsets* (classes) of customers.

The UL and GM bounds by themselves are very natural to consider and have in fact been specifically proposed in a special context by Kamoun and Kleinrock [6]. Moreover, it is evident that the UL and GM bounds lead to a coordinate-convex sharing policy, so that the resulting resource-sharing model has a product-form steady-state distribution. The main contribution of this invention is to develop an effective method for calculating blocking probabilities in this model, so that the various problems faced by the resource-provider can be solved.

It is evident that the UL bounds can easily be enforced provided that the resource provider keeps track of the number of requests from each customer in service, but it not so evident that the GM bounds can easily be enforced. Nevertheless, an *efficient scheme for enforcing both the GM and UL bounds* is developed.

The UL and GM bounds either may be provided directly by the customer or may be determined by the resource provider to satisfy other requirements. It is assumed that the customer provides his blocking requirement and a characterization of his desired traffic. The request blocking requirement is the maximum allowed request blocking probability, assuming that all customers submit requests according to their negotiated traffic parameters.

A customer's traffic is characterized by the request arrival rate, the average request holding time, and the number of units needed on each resource. In addition, a *burstiness parameter*, which describes the variability of the customer's request stream, is also allowed. The standard assumption is a Poisson arrival process. The burstiness parameter accounts for arrival processes substantially more or less bursty than a Poisson process. The customer requests can even have state-dependent arrival rates or batch arrivals.

Customer grades of service can also be specified in another way, in particular, via *conditional blocking requirements*, which are blocking requirements conditional on the other customers being in some pattern of overload. For one example, there can be a conditional blocking requirement given that any one other customer is in arbitrary overload. For another example, there can be a conditional blocking requirement given that all other customers are in 10% overload, i.e., given that each other customer is submitting traffic at a rate 10% above its negotiated rate. The UL and GM bounds are used to ensure that the conditional blocking requirements are met.

2. An Efficient Blocking Probability Computer

In order to efficiently manage the shared resources with the proposed grades of service, the invention contains new ways to approximately and exactly solve the natural resource-sharing model with these grades of service in effect. This model is solved by the "blocking probability computer."

As indicated in the background section, the resource-sharing model with UL and GM bounds has a product-form steady-state distribution. To surmount the difficulty provided by the extra constraints, a new algorithm is introduced. The algorithm exploits the known fact that the desired blocking probabilities can be directly expressed in terms of normalization constants (or partition function) appearing in the product-form steady-state distribution. The essential step of the algorithm is to compute the normalization constants. The normalizing constants are computed by first constructing the generating function (or z-transform) of the normalizing constant and then numerically inverting the generating function. This has not been done before.

Since numerical inversion of the generating function can be difficult, the algorithm includes several important steps to facilitate the inversion. The main difficulty is that the required computation grows exponentially in the dimension of the generating function. The dimension of the generating function is in turn equal to the number of resources plus the number of separate linear constraints. Since the UL and GM bounds typically correspond to a large number of linear constraints, it is necessary to simplify the problem even for a single resource.

The first step is to use a *normal approximation scheme* to approximately determine the load on each resource. This normal approximation scheme extends previous work by Whitt [10] and Dziong and Roberts [5]. This preliminary approximate analysis, determines if there are some resources that are so lightly loaded that they provide essentially no constraint, so that they can be eliminated from consideration. These very lightly loaded resources are eliminated from the model before constructing the generating function.

After performing the preliminary analysis with the normal approximation, the generating function for the model with all remaining resources is formed. An explicit expression for this generating function has been determined as a function of the customer parameters.

The next step is to look for another way to reduce the effective dimension of the generating function. A *conditional decomposition scheme* is used to determine a good order for inverting the variables of the multidimensional generating function. The conditional decomposition scheme often can significantly reduce the effective dimension of the generating function. Indeed, this conditional decomposition step always works with the UL and GM bounds to reduce the dimension to at most two more than the dimension with the complete-sharing policy. If there is only a single resource, then the resulting model is almost always solvable.

If there is a substantial number of resources in the model, then the model may still not be solvable after the normal approximation and conditional decomposition steps. If the model is not yet solvable, then a *reduced-load fixed-point approximation scheme* is used to approximately solve the model. While the general idea of reduced-load fixed-point approximation schemes is part of the current state of the art, e.g., see Chung and Ross [3], Dziong and Roberts [5], Kelly [8] and Whitt [11], this version is new because it uses as a subroutine the new algorithm to exactly solve the single-resource submodels having UL and GM bounds. The new algorithm may also be used to exactly solve models involving subsets of more than a single resource. In either case, the basic inversion algorithm is an essential component of the reduced-load approximation scheme.

After the three problem-reduction steps are completed, it remains to actually perform the numerical inversion to calculate the blocking probabilities. This is either done directly, if the model is directly solvable, or indirectly as part of a subroutine within the reduced-load fixed-point approximation scheme. The inversion itself involves several key new steps. The inversion exploits the Fourier-series method, as presented in Abate and Whitt [1] and Choudhury, Lucantoni and Whitt [2]. A key new ingredient is an effective scaling algorithm especially tailored to the resource-sharing model. The inversion also exploits truncation and an efficient treatment of multiplicities to obtain significant computational speedup.

With the innovations just described, the blocking probability computer is able to determine whether proposed grades of service can be provided to a given set of customers by resources with given capacities. Even without the UL and GM bounds, the blocking probability computer is able to solve much larger resource-sharing models than could be solved previously.

The invention also includes ways to determine the *costs* of the grades of service where cost here refers to the capacity required to provide service. Different notions of this cost are identified and ways to estimate these costs are developed, exploiting the blocking probability computer.

It remains to describe how the grades of service and the blocking probability computer are used to actually manage the shared resource. These aspects of the invention are summarized next.

3. Real-Time Admission Control

The invention exploits the blocking probability computer to determine in real time whether each new prospective customer can be admitted with a desired grade of service. The first step is to determine the desired grade of service for the prospective customer. This is done by first telling the customer the available service options and then receiving input from the customer. This step yields proposed customer traffic parameters and UL and GM bounds. The resource provider then invokes the blocking probability computer to solve the model containing both the prospective customer and all existing customers. The solution enables the resource provider to determine if all blocking requirements (conditional and unconditional) can be met. If all blocking requirements can be met, then the resource provider admits the new customer with the desired grade of service. If all blocking requirements cannot be met, then the resource provider uses a search algorithm and the blocking probability computer to find alternative requirements at a lower grade of service that are feasible. When the resource provider finds feasible lower grades of service, these are offered to the customer. If the customer accepts a lower grade of service, then the customer is admitted with this lower grade of service. Otherwise the customer is not admitted.

4. Capacity Adjustment

The invention also exploits the blocking probability computer to determine appropriate capacity adjustments to make to meet changes in customer demand. Given a new set of prospective customers with specified resources and grades of service, the blocking probability computer is used to find appropriate resource capacities. A normal approximation scheme is first used to find an upper bound on the feasible capacities. Then the capacities are steadily decreased with the UL and GM bounds changed as necessary. The blocking probability computer is used to evaluate each candidate. The search yields feasible capacities that are as small as can be found by this method.

5. Response to Resource Failure

The invention exploits the blocking probability computer to determine an effective response to resource failure. Available methods are used to determine how customers who were using one of the failed resources can have their demand satisfied on the remaining resources. The blocking probability computer is then used to see if the new arrangement is feasible. If it is not feasible, then the resource provider considers capacity expansions. if capacity expansions are not possible, then the UL and GM bounds are used to provide appropriate protection for the original and diverted customers.

References

[1] J. Abate and W. Whitt, "The Fourier-Series Method for Inverting Transforms of Probability Distributions," *Queueing Systems*, vol. 10, pp. 5–88, 1992.

[2] 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, pp. 719–740, 1994.

[3] S. P. Chung and K. Ross, "Reduced Load Approximations for Multi-Rate Loss Networks," *IEEE Transactions on Communications*, vol. 41, pp. 1222–1231, 1993.

[4] L. E. N. Delbrouck, "On the Steady-State Distribution in a Service Facility Carrying Mixtures of Traffic with Different Peakedness Factors and Capacity Requirements," *IEEE Transactions on Communications*, vol. 31, pp. 1209–1211, 1983.

[5] Z. Dziong and J. W. Roberts, "Congestion Probabilities in a Circuit-Switched Integrated Services Network," *Performance Evaluations*, vol. 7, pp. 267–284, 1987.

[6] 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. COM-28, pp. 992–1003, 1980.

[7] J. S. Kaufman, "Blocking in a Shared Resource Environment," *IEEE Transactions on Communications*, vol. COM-29, pp. 1474–1481, 1981.

[8] F. P. Kelly, "Loss Networks," *Annals of Applied Probability*, vol. 1, pp. 319–378, 1991.

[9] J. W. Roberts, *Performance Evaluation and Design of Multiservice Networks*, COST 224 Final Report, Commission of the European Communities, Luxembourg, 1992.

[10] W. Whitt, "Heavy-Traffic Approximations for Service Systems with Blocking," *AT&T Bell Laboratories Technical Journal*, vol. 63, pp. 689–708, 1984.

[11] W. Whitt, "Blocking When Service is Required from Several Facilities Simultaneously," *AT&T Technical Journal*, vol. 64, pp. 1807–1856, 1985.