

# Connection Admission Control for Differentiating Priority Traffic on Public Networks

Cory C. Beard

University of Missouri-Columbia/Kansas City, Department of Electrical and Computer Engineering  
5100 Rockhill Road, Kansas City, MO 64110

Victor S. Frost

University of Kansas, Information and Telecommunication Technology Center  
Department of Electrical Engineering and Computer Science  
Lawrence, KS 66045, E-mail: frost@ittc.ukans.edu

**Abstract**-The public network has traditionally been unable to adequately deal with defense and disaster recovery communications, because overloads that occur during crises cause degraded resource access to all users, no matter how important. For public broadband networks to be effective for defense and disaster recovery multimedia communications, they must dynamically recognize some connections as having greater importance than others and allocate resources accordingly. A new approach to connection admission control is proposed that uses an upper limit policy to optimize the admission of connections based on the weighted sum of blocking across traffic classes. This optimization approach can be used for arbitrarily large networks and numbers of traffic classes and results in a very simple algorithm that could be implemented on standard network hardware. This work is also the first to demonstrate that the use of an upper limit policy is superior to traditional approaches of adding extra capacity or partitioning capacity, both in the amount of resources required and in sensitivity to load variations. An upper limit policy is also shown to be much faster to implement when a large overload occurs from a disaster event.

## I. INTRODUCTION

The expansion in the public data network over the past decade has been astounding, both in terms of its widespread deployment and in the usefulness of its applications, most notably with the world wide web. Although having a potentially profound impact for defense and disaster recovery operations [1], however, the public network remains a virtually unusable resource [2]. Current public network resource allocation mechanisms do not prioritize the way they allocate resources, instead working on a first come-first served basis. When loads on public networks reach up to five times normal during an emergency [3], important traffic receives equally poor service as low priority traffic. In a report by the National Research Council [4], this problem was referred to by emergency management experts as the need to give "emergency lane" access to resources.

Even though development of mechanisms to support differentiated or guaranteed quality of service (QoS) promises predictable service to connections once they are established, the question of how and to whom to give access to those resources has not been fully addressed. Emergency response requires access to all available communication resources. If not available from public networks, communications resources must be obtained through use of higher cost dedicated networks.

To meet the demands of defense and disaster recovery users, connection admission control (CAC) functions must be able to dynamically designate some connections as having greater importance than others and then provide access to

resources accordingly. Those with greater importance may be those which generate greater revenue, but also may be those that deal with emergencies or natural disasters. Related research [5] addressed the question of how to determine which traffic should be given priority; a ticket server architecture was proposed to issue tickets to important connections for use when seeking connection admission.

The question of how to allocate resources to important connections once they attempt to use those tickets is addressed here. Of particular interest are times of network stress when connection requests must be denied (i.e., blocked) to preserve quality of service for other connections. Connection requests should also be blocked if admission of the current request might lessen the possibility of a subsequent, more important request being admitted.

One approach to the problem would be to use traditional telecommunications network techniques. These have either been to overbuild networks with excess capacity to keep requests from virtually ever being denied or to partition resources into separate pools that cannot be used by others. A variety of newer approaches have recently been proposed to deploy new communications resources to a stressed network as needed, but these are really only a variation on the excess capacity approach. The approach proposed here is to use an *upper limit* (UL) policy for connection admission that uses resources currently available and sets upper limits on the amount of resources that can be used for each prioritized class. While the concept of an upper limit policy is not new, this research makes the following contributions.

- It provides the first detailed comparison of the implementation benefits of an upper limit policy versus excess capacity or partitioned capacity approaches. Of particular interest is the amount of network capacity needed for each approach and each approach's sensitivity to the large load variations that can occur during major disaster events. For typical scenarios, excess capacity approaches are found to require more than double the capacity of a UL approach; partitioned capacity can cause 50% more blocking than for a UL approach.
- An approach is proposed to directly control blocking that optimizes upper limit policies based on a weighted sum of blocking across traffic classes.
- A simple linear optimization formulation and a simple algorithm are developed to optimize upper limit policies. It is developed from an approximation for asymptotically large networks in overloaded conditions. An arbitrarily large number of traffic classes can be used.

The next section defines the scope of the problem.

Following that is a discussion of related work on connection admission policies and approximations of blocking for those policies. Next, in Section IV, is the development of the upper limit policy optimization algorithm. Sections V and VI then compare optimal upper limit policies to excess capacity and partitioned capacity resource allocation approaches. Section VII then provides a detailed example of how an upper limit policy might be used during stages of a disaster response effort, and Section VIII provides the conclusion.

## II. PROBLEM STATEMENT

Two problems to be considered are the following.

1. Is dynamic prioritization of resources really beneficial? A dynamic prioritization approach requires both an architecture to determine which connections are more important and algorithms to allocate resources. Would more traditional methods of allocating resources be just as useful and simpler to implement? The traditional approaches are to either have enough excess capacity (or newly deployed capacity) to keep blocking low or to partition resources so high priority traffic has its own dedicated pool. These are illustrated in Fig. 1.
2. If resources are dynamically prioritized, how would connection admission control (CAC) functions decide which connections to admit? Could algorithms be simple enough for use on standard networking hardware?

The approach to addressing these questions will be to first answer the second question. A simple, efficient CAC process is developed that provides preferred access to important connections by optimizing the maximum number of connections allowed per class. Then the first question will be addressed. The proposed CAC algorithm will be shown to provide much better utilization of resources than using excess capacity and much better robustness to load variations than partitioning resources.

The basic context for the problem lies in the application of stressed network conditions to the problem of *loss networks*. In a loss network, requests for connections are either accepted or blocked; no queueing of requests occurs. To prioritize the allocation of resources for important connections, the objective is to minimize the weighted sum of blocking,

$$W_B = \text{weighted sum of blocking} = \sum_{r=1}^R w_r P_{B_r}, \quad (1)$$

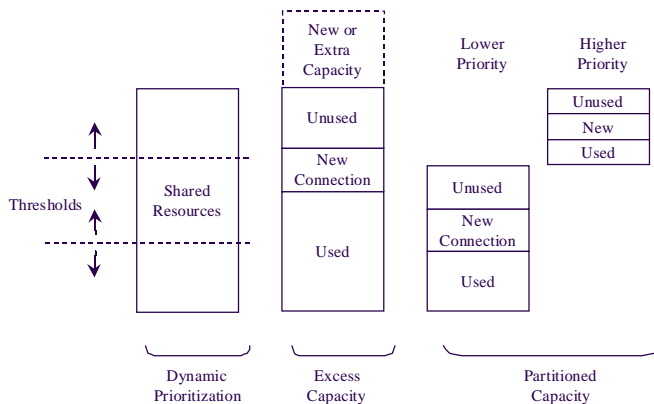


Fig. 1 - Resource Allocation Alternatives

where

$R$  = number of classes

$w_r$  = weight for class  $r$

$P_{B_r}$  = probability of blocking for class  $r$ .

While others have proposed optimization based on maximizing revenue or utilization [6, 7], this is the first time a weighted blocking metric has been used to directly control and monitor blocking and also set bounds on blocking for particular classes. A set of weights reflects the relative cost of blocking for each class. A policy can be formulated from these weights to minimize  $W_B$  and give direct feedback about the blocking experienced by different classes of users. This weighted blocking criteria is also used in later sections as a basis of comparison between resource allocation approaches.

The network is assumed to use connection-oriented resource allocation to provide differentiated quality of service levels, and could apply to work being done many areas (ATM, TCP/IP, etc.). An arbitrary number of traffic classes is allowed, with each class defined by an importance level and the amount of resources used by each connection. Connection requests arrive according to an independent and identically distributed Markov process. Service times, however, are generally distributed [8]. Estimates for the current overall load and load per class are provided by a ticket server architecture [5] by tracking the frequency of ticket requests. The analysis looks at the case of one communication link and one resource (bandwidth), but the approach can be extended to include other multimedia resources (buffers, time slots, etc.) and a network of links.

## III. RELATED WORK

In the most general case of resource allocation, all connections are admitted simply if resources are available at the time a connection is requested. This is commonly called a *complete sharing* (CS) admission policy where the only constraint on the system is the overall system capacity,  $C$ . In a CS policy, connections that request fewer resource units are more likely to be admitted. This policy also does not consider the importance of a connection when resources are allocated.

### A. Types of Policies

Other policies have been derived to provide a more equitable balance between users or to provide optimized access to resources. Ross [9] provides extensive discussion of different approaches that have been taken. Some have derived optimal policies [10, 11, 12, 13, 14, 15, 16]. To implement optimal policies, however, a detailed accounting must be made of every allowable state and state transition, which is impractical for networks of even modest size. Therefore, a set of non-optimal, heuristic policies have been developed that are simpler to implement and provide a more intuitive understanding of how resources are managed. In a *complete partitioning* (CP) policy, every class of traffic is allocated a set of resources that can only be used by that class. A *trunk reservation* (TR) policy says that class  $i$  may use resources in a network up until the point that only  $r_i$  units remain unused. A *guaranteed minimum* (GM) policy [17, 18] gives each class their own small partition of resources. Once

used up, classes can then attempt to use resources from a shared pool that all classes use. And finally, an *upper limit* (UL) policy, introduced in [17], places upper limits on the numbers of connections possible from each class to ensure that no one class can dominate the use of resources.

Several comparisons have been made between heuristic policies and with the optimal policy. The upper limit policy was found to be optimal for a subset of two class policies [12] and a subset of policies for asymptotically large links [7]. CP, GM, UL, and TR policies were found to outperform the CS policy only when significant differences in blocking probabilities between classes were required [19]. UL and GM policies can significantly outperform TR policies, especially when trying to control blocking performance [18].

The research here proceeds with further development of the upper limit policy. A UL policy is competitive with TR and optimal policies when blocking performance is to be controlled. More importantly, the simplicity of its definition provided opportunities for developing the simple optimization algorithms that are presented in the next section. Fig. 2 illustrates an upper limit policy for two classes of traffic. It shows a linear bound on the number of connections that the CS policy imposes, and the additional thresholds for each class imposed by the upper limit policy. A valid upper limit policy need not implement a threshold for every class.

### B. Computational Methods

Most of the work on computing blocking for CAC policies has centered on the Erlang loss function, which provides the ability to exactly compute blocking for different policies [7] [10-16], but can only be used when networks are small (less than 1000 units of capacity). In detailed work on upper limit policies, [17, 18] provide a numerical inversion method using moment generating functions to find exact blocking probabilities for UL and GM policies. They state, however, that “the numerical inversion algorithm can also have high complexity ... the current upper limit on the dimension (number of classes) amenable for computation is about five.” [18] Optimal UL threshold parameters have been found in [18] and [19], but only using heuristic search algorithms. The goal here, however, is to find the optimal parameters using direct optimization with no practical limit on the number of classes or the size of the network.

Approximations for the Erlang loss function as networks asymptotically grow in size are particularly useful in this regard. This is a reasonable assumption, since an OC-48 link can support over 30,000 64 kilobit/sec voice channels, for example. An important work was produced by Kelly [20] and then later expanded by Hunt and Kelly [21]. Kelly approximated blocking probabilities from the expected value of the number of connections in progress in a network. Kelly’s theory was used as the basis for [22, 23, 24, 25, 26, 27], and also our work on upper limit policies that is described in more detail later. No work has been done to date on upper limit policies in overloaded conditions, where load and capacity tend to infinity at a constant ratio, but with load greater than capacity. Thus the contribution of this research is to asymptotically approximate blocking for upper limit policies in overloaded conditions and to directly optimize upper limit policies rather than use heuristic searches.

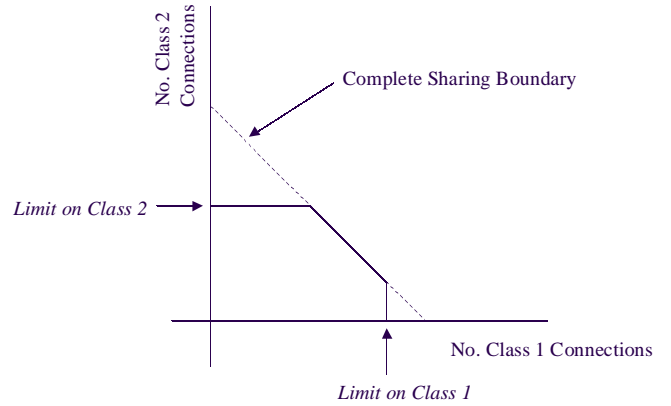


Fig. 2 Illustration of the Upper Limit Policy

## IV. DERIVATION OF THE UPPER LIMIT POLICY

This section provides a new derivation of an optimal upper limit control policy from a set of classes each defined by an importance level and a resource requirement. The upper limit policy is optimized to minimize the weighted sum of blocking, where weights signify desired relative blocking performance. The system under investigation has a single resource (bandwidth), a single link, and an arbitrary number of classes. Load and capacity asymptotically approach infinity proportionally at a constant ratio of load to capacity greater than 1 (i.e., an overloaded condition).

### A. Blocking Probabilities for an Upper Limit Policy

Kelly’s formulation for asymptotically large networks [20] is based on a network where each class of traffic uses an integer number of resources along a path. The analysis assumes that call holding periods are generally distributed with unit mean [20]. Each class of traffic is defined by the route each connection takes and the amount of resources it uses on each link. A class of traffic is limited in the number of simultaneous connections it can have by complete sharing policies on each link it traverses, with capacity constraint  $C_j$  for link  $j$ . The constraints for all links on the network are

$$\mathbf{A}\mathbf{n} \leq \mathbf{C} \quad (2)$$

where row  $j$  of  $\mathbf{A}$  defines the CS constraint for link  $j$ . The vector  $\mathbf{n}$  is the number of connections in progress per class, and  $\mathbf{C}$  is the vector of link capacities.

Kelly [20] then proceeds to find the most likely state,  $\mathbf{n}$ , and shows that the normalized expected value of the number of connections in the system asymptotically converges to the most likely state [20]. Using  $\lambda_r$  as the arrival rate per class, the blocking per class can therefore be found from finding the set of  $y_j$ ’s that optimize the minimization problem

$$\min \sum_{r=1}^R \lambda_r e^{-\sum_{j=1}^J y_j A_{jr}} + \sum_{j=1}^J C_j y_j \quad (3)$$

subject to  $y_j \geq 0$ .

Blocking is then

$$P_{B_r} = 1 - \prod_{j=1}^J e^{-y_j A_{jr}} \quad (4)$$

The new work here is to reformulate Kelly’s problem

using a single link with an upper limit policy being used to impose upper limit thresholds on each class. To use the results from [20], the form of the constraints,  $\mathbf{A}\mathbf{n} \leq \mathbf{C}$ , are changed. Instead of defining  $\mathbf{A}$  by CS constraints per link,  $\mathbf{A}$  is formulated to include the constraints for the upper limit policy, which are

$$\sum_{r=1}^R b_r n_r \leq C \quad (5)$$

for the overall capacity of the link and

$$b_r n_r \leq L_r, \quad (6)$$

where  $n_r$  is the number of connections in progress for class  $r$ ,  $b_r$  is the number of resource units used by each connection in class  $r$ , and  $L_r$  is the upper limit on class  $r$ . The form of the matrices then becomes

$$\mathbf{A}\mathbf{n} \leq \mathbf{C} \quad (7)$$

$$\begin{bmatrix} b_1 & b_2 & b_3 & \cdots & b_{R-1} & b_R \\ b_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & b_2 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & b_{R-1} & 0 \\ 0 & 0 & 0 & \cdots & 0 & b_R \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ \vdots \\ n_R \end{bmatrix} \leq \begin{bmatrix} C \\ L_1 \\ L_2 \\ L_3 \\ \vdots \\ L_R \end{bmatrix}.$$

The result is Kelly's same optimization problem over a state space that is constrained by  $\mathbf{A}\mathbf{n} \leq \mathbf{C}$ , now with  $\mathbf{A}$  defined by thresholds from an upper limit policy.

Since  $\mathbf{A}$  is a sparse matrix, solving the minimization problem of equation (3) results in blocking for class  $r$  of

$$P_{B_r} = 1 - \frac{L_r}{b_r \lambda_r}, \quad (8)$$

subject to

$$0 \leq L_r \leq b_r \lambda_r \quad (9)$$

$$\text{and } \sum_{r=1}^R L_r = C.$$

Since  $\lambda_r$  and  $b_r$  are constants, the blocking probability of a class  $r$  connection is a linear function of the upper limit threshold for that class,  $L_r$ . The interaction between blocking for different classes is through  $\sum_{r=1}^R L_r = C$ , which indicates that for every increase in  $L_r$  to lower blocking for one class, one or more other classes must decrease their  $L_r$  and experience an increase in blocking.

### B. Optimal Upper Limit Thresholds

With (8), an optimal upper limit policy can be derived to minimize the weighted sum of blocking. Starting with the weighted blocking metric in (1), the following linear program is formed to optimize upper limit thresholds

$$\min \left( - \sum_{r=1}^R a_r L_r \right)$$

subject to

$$\begin{aligned} L_r + s_{r,1} &= L_{r,\max} = b_r \lambda_r (1 - P_{B_{r,\min}}) \\ L_r - s_{r,2} &= L_{r,\min} = b_r \lambda_r (1 - P_{B_{r,\max}}) \end{aligned} \quad (10)$$

$$\sum_{r=1}^R L_r = C.$$

$$L_r, s_{r,1}, s_{r,2} \geq 0.$$

where  $a_r$  is the ratio of weight to load for class  $r$ , found from

$$v_r = b_r \lambda_r = \text{load for class } r$$

$$a_r = \frac{w_r}{b_r \lambda_r} = \frac{w_r}{v_r}. \quad (11)$$

$L_{r,\max}$  and  $L_{r,\min}$  are derived from maximum and minimum blocking values that can be defined for each class.

Optimization results for an example with two classes are provided in Figures 3 and 4. The link is overloaded at a ratio of overall load ( $v_{tot}$ , the sum of  $v_r$ ) to capacity of 2. The weight of the high priority load (class 1) is ten times that of class 2 and its load is 1/10 that of class 2. The plots show how the blocking probabilities for each class change as the upper limit on class 2 ( $L_2$ ), changes. The optimal value is  $L_{2,opt}=818$ .

Fig. 3 shows approximate blocking probabilities compared to the upper limit for class 2; as  $L_2$  decreases, blocking for class 2 increases gradually while blocking for class 1 drops sharply, because high priority load is smaller. Small changes in  $L_2$  make a bigger impact on that class. The optimization process makes the blocking for class 1 go to approximately zero. Fig. 3 also shows actual blocking using Erlang's formula. Most notable is the deviation between approximate and actual values for blocking for class 1 in the area of  $L_2=800$ . For all other areas until  $L_2$  approaches  $C=1000$ , approximate values are close to actual values. The

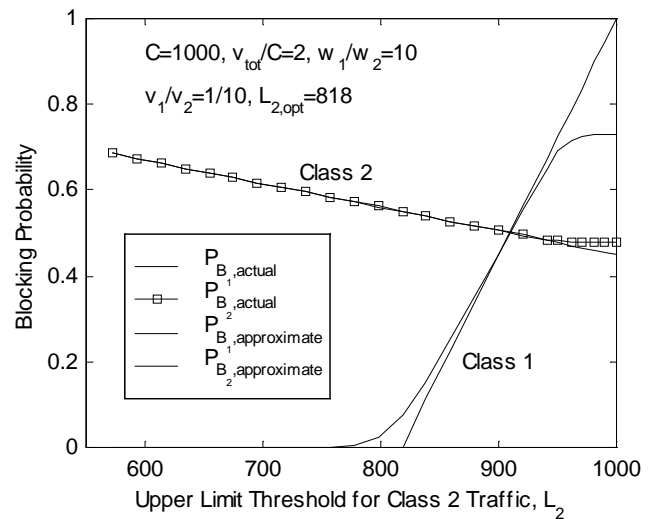


Fig. 3 Blocking Variation as the Upper Limit on Class 2 ( $L_2$ ) Changes

error in the approximation is on the order of  $1/\sqrt{C}$ , where  $C$  is the overall capacity of the link. In practice,  $L_2$  might be set lower than  $L_{2,opt}=818$ , (e.g.,  $L_2=780$ ), which would make actual blocking for class 1 nearer to zero and blocking for class 2 slightly higher.

Fig. 4 shows how the weighted sum of blocking,  $W_B$ , changes with  $L_2$ . By decreasing  $L_2$  to decrease the blocking on class 1,  $W_B$  drops off sharply, since class 1 is weighted more highly. The gradual increase in blocking for class 2 does not significantly affect  $W_B$ .

The significance of these results is seen in the reduction in  $W_B$  that occurs. By implementing the UL policy,  $W_B$  is reduced from 0.7 to 0.05, only 7% of the weighted blocking as without the UL policy (i.e., a CS policy). This is because blocking for high priority traffic goes from 0.75 to approximately 0. The upper limit policy caused the blocking for the low priority traffic to rise from 0.48 to 0.52, certainly a reasonable penalty.

### C. Upper Limit Threshold Optimization Algorithm

The linear program given in (10) can also be formulated as the following algorithm.

1. Compute all  $a_r = \frac{w_r}{v_r}$  and sort in descending order.
2. Allocate the minimum  $L_r$  to each class.
3. If  $\sum_{r=1}^R L_{r,allocated} > C$ , stop. No feasible solution is possible for this set of minimum  $L_r$ 's. Constraints on the minimum  $L_r$ 's come from  $L_{r,min} = b_r \lambda_r (1 - P_{B_{r,max}})$ , so maximum blocking probabilities must be higher or loads  $(\lambda_r b_r)$  lower for a feasible solution to exist.
4. Find the remainder of  $C$  that can still be allocated,

$$C_{remaining} = C - \sum_{r=1}^R L_{r,allocated} .$$

5. Find the class,  $r$ , which has the largest  $a_r$ .
6. Form a new  $L_r$  for that class by either allocating all of

$C_{remaining}$  or increasing  $L_r$  to its upper limit, whichever would increase  $L_r$  the least, according to

$$L_r = L_{r,allocated} + \min(C_{remaining}, L_{r,max} - L_{r,allocated}) .$$

7. Update  $C_{remaining}$ .
8. If  $C_{remaining} = 0$ , stop. The set of  $L_r$ 's is the optimal solution.
9. If  $C_{remaining} > 0$ , move down the list of  $a_r$ 's to the next class. If no more classes exist, stop. No feasible solution is possible since the sum of the maximum  $L_r$ 's is less than  $C$ . Constraints on maximum  $L_r$ 's come from  $L_{r,max} = b_r \lambda_r (1 - P_{B_{r,min}})$ , so minimum blocking probabilities must be lower or loads  $(\lambda_r b_r)$  higher for a feasible solution to exist.
10. Otherwise, go back to step 6.

The above new algorithm is simple enough to implement on standard network hardware. Proof that this algorithm produces the optimal solution comes from the fact that no modifications to the set of  $L_r$ 's would produce a better  $W_B$ . Those classes where an increase in  $L_r$  would improve  $W_B$  are already at their maximum, while classes where a decrease in  $L_r$  would improve  $W_B$  are already at their minimum.

### V. COMPARISON OF EXCESS CAPACITY AND UPPER LIMIT POLICIES

An asymptotic approximation that allows optimization of upper limit thresholds has been found. While it has advantages over other resource allocation policies and is efficient to implement, it still must be considered against traditional resource management approaches. The first alternative to consider is using excess capacity in the network to keep blocking low, either by overbuilding or by deploying new capacity as the need arises. A second alternative, complete partitioning, will be considered in the next section. Both alternatives were illustrated in Fig. 1.

#### A. Numerical Comparisons of CS and UL

The key issue is not whether a CS policy could be implemented to provide the same weighted blocking as a UL policy, but rather *how much* capacity would be required to make this happen. Fig. 5 shows how much CS capacity is needed to provide comparable weighted blocking to a UL policy at various overloads. A linear relationship is seen in Fig. 5, indicating that for every increase in load, a proportional amount of new CS capacity would have to be installed to keep  $W_B$  the same as a UL policy. The slope of the line, which we call the *increment ratio*, in this case equals 0.63. A 100% increase in load would require 63% more new capacity; 160% more load would require 100% more capacity. After disasters, 5 times normal load is possible [3].

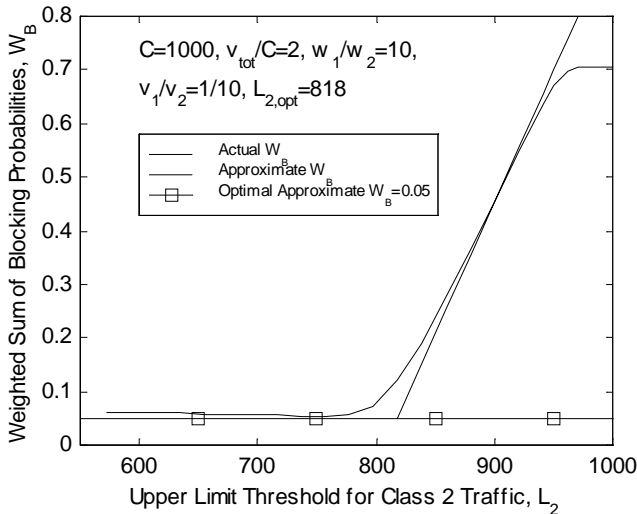
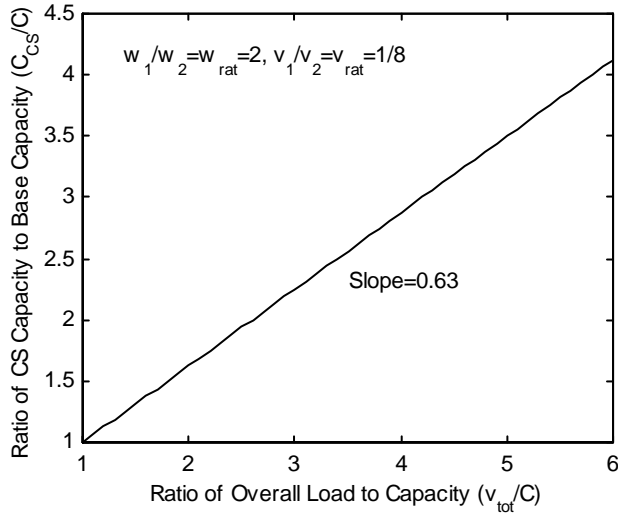


Fig. 4 Weighted Blocking Variation ( $W_B$ ) as the Upper Limit on Class 2 ( $L_2$ ) Changes

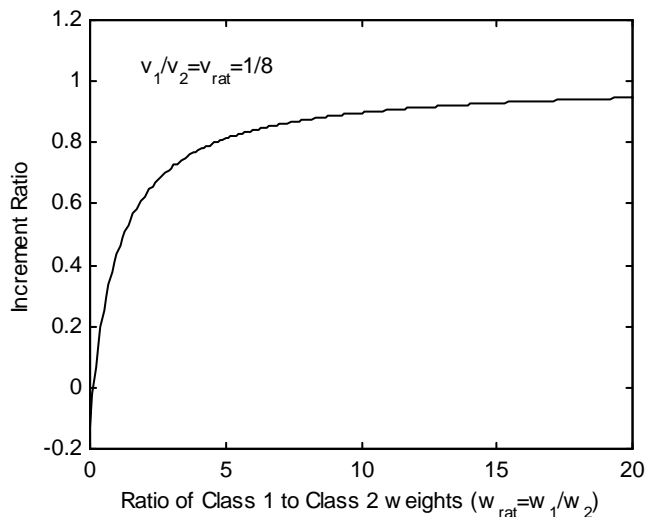


**Fig. 5 Extra CS Capacity Needed versus a UL Policy in Overloaded Conditions**

Fig. 6 shows how the increment ratio changes as the ratio between weights,  $w_{rat} = w_1/w_2$ , is varied. The relationship between  $w_{rat}$  and the increment ratio is nonlinear and asymptotically approaches 1. This asymptotic limit means that a CS policy will never need to increase capacity more than the amount load has increased. Typical values of  $w_{rat}$  would be larger than 3, resulting in an increment ratio of at least 0.7. A 150% increase in load (typical during disasters) would require CS capacity to at least be doubled.

### B. Comparison of CS and UL on Implementation Time Scales

Another potentially more important issue in the comparison of CS and UL policies is the time scales on which they can be implemented. If capabilities for a UL policy are already installed in network hardware, a UL policy can be implemented within minutes of a major overload. The only delay would be to have time to assess new load levels. For the CS policy, however, to deploy new capacity when it was needed, it would take at least several hours and possibly days for this to occur. The use of a CS policy would result in



**Fig. 6 Increment Ratio vs.  $w_{rat}$**

extremely high blocking for several hours at the beginning of a major event when resources are needed most. Section VII provides an example that illustrates this. It should also be noted that the overloads that occur as a result of a major event typically are limited to the first day or two of an event [28]. If new CS capacity is not deployed soon enough, it might miss the overload period completely.

### C. Summary of the Comparison of CS and UL

Following summarizes the comparison of UL and CS policies.

1. CS policies use considerably more capacity than UL policies. Extra capacity must be deployed at a rate 0.7 to 1.0 times the amount of the extra load from the disaster.
2. Installation of new capacity takes much longer than instituting an upper limit policy.
3. Typical load surges for a disaster last one or two days, so if installation of new capacity takes too long, it may not provide any benefit during the peak loading periods.

## VI. COMPARISON OF FIXED PARTITIONS AND UPPER LIMIT POLICIES

The other key comparison that must be made is between the upper limit policy and complete partitioning policies. Complete partitioning has traditionally been the approach used to provide disaster response communications [29] [30]. This approach is attractive because it separates resources from the general public. The public network is usually unable to adequately deal with defense and disaster recovery communications because it becomes so overloaded that access to resources is virtually impossible. A separate set of resources is immune to such overloads.

The problems with such an approach are twofold, however, as exemplified in the following quote.

Radio systems designed and used by emergency management agencies appear to be virtually unused on a day-to-day basis, yet when a major event occurs, these same systems are inadequate for meeting the need to communicate. [29]

Thus the two problems are

1. Wasted, unused resources on a day-to-day basis.
2. Not enough resource access (i.e., high blocking) during major events because of large load increases. CP keeps other classes from using the high priority partition, but also keeps high priority traffic confined only to the resources in that partition.

Clearly the use of a UL policy can increase resource utilization to address the first problem. The UL policy does not keep any resources unused, but allows use by all users.

The UL policy can address the second problem in two ways. First, if an existing resource management system using CP cannot dynamically adjust the resources allocated to each class, UL is better simply because it can dynamically adjust. If the CP system can dynamically adjust, however, the UL policy is still better because, as stated in Section III, a valid upper limit policy need not implement thresholds for every class. For those classes where  $L_r$  is set to  $L_{r,max}$ , no threshold needs to be implemented. For those classes, the UL policy is, therefore, less sensitive to load changes since the traffic is not constrained by an upper limit.

Fig. 7 shows a comparison of class 1 (high priority) blocking for the UL and CS policies that starts at  $v_1/v_2=0.1$  and  $w_1/w_2=2$ . The ratio of class 1 load to normal class 1 load is then varied from 1 (equal to its normal load) to 10 times its normal load. Blocking probabilities start below 0.1 and then grow sharply as load increases. Until the load ratio reaches 1.5, the two policies provide about equal blocking. Once the load ratio increases beyond 1.5, however, blocking for the CP policy is up to 50% higher than for the UL policy. At any load ratio, if the thresholds can be recomputed for either CP or UL, blocking can be made much lower, but if not, load fluctuations will affect CP performance much more than UL performance.

## VII. EXAMPLE

To illustrate the optimal upper limit policy methods proposed here, consider an analysis of an example disaster event where disaster response activities and network capabilities dynamically move through a series of stages. An OC-12 link is used, and load is from five classes of traffic listed in Table 1. Three high importance classes have a weight of 0.3; one medium importance class has a weight of 0.08; and one low importance class has a weight of 0.02. One high importance class uses a 1.28 Mbs video link, while all other classes use 64 kbs for audio or 128 kbs for low rate video or general Internet access. Most of the load is in Class 3, a high importance voice class (to simulate E-911 connections) and Class 5, the low priority class. During normal conditions, overall load is 90% of capacity and a CS policy is used. Blocking normally is on the order of  $10^{-6}$  for all classes.

Now consider the state of the network as it goes through the following stages.

Stage 1 - A major earthquake occurs and network load doubles [3]. With a CS policy being used, blocking will increase considerably. Resulting blocking probabilities are given in Table 1. The bias of the CS policy toward lower blocking for lower bandwidth traffic can be seen; high bandwidth video connections are unusable. High

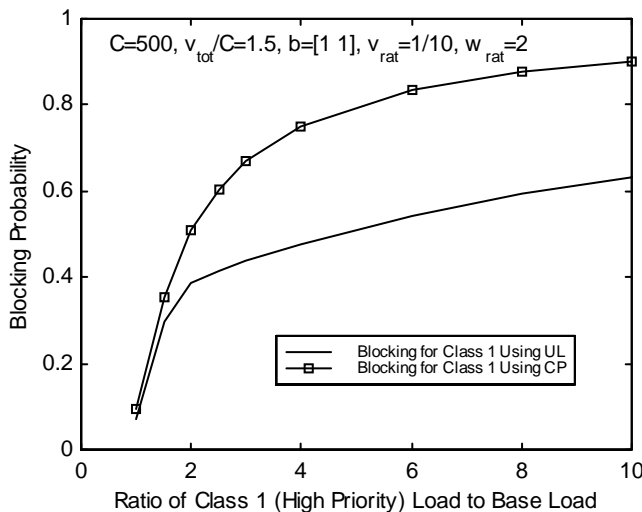


Fig. 7 Changes To High Priority Blocking For UL Vs. CP

priority traffic uses higher bandwidth services, so blocking for high priority traffic overall is poor.

For stage 2, two options are possible.

Stage 2a - After 5 minutes the network is able to recognize the large load and implement an optimal upper limit policy. Assuming software capabilities have already been installed, the upper limit policy is implemented according to Section IV, with blocking for all classes constrained to be less than 0.9. Using this UL policy, blocking for classes 1 through 4 goes to approximately zero, while blocking for class 5 goes to 0.74 (up from 0.36 when using CS). This results in a improvement in the sum of weighted blocking to  $W_B=0.0148$ , only 2% of the  $W_B=0.637$  for the CS policy, which means 50 times as many important connections are admitted.

Stage 2b - If an upper limit policy is not implemented, after 6 hours new capacity can be installed. To have the same weighted sum of blocking as the UL policy from Stage 2a, capacity must be increased from 8064 to 14650 units (64 kbs is one unit), an 82% increase in capacity. Resulting blocking probabilities for the lower bandwidth classes are on the order of  $10^{-3}$ . The high importance video class (class 1) has blocking of 0.0424, which might still be too high, since it is high importance traffic.

This example illustrates that a CS policy is only useful in underloaded conditions (i.e., load less than capacity). Once loads are above capacity, CS blocking increases considerably and lower bandwidth classes are given lower blocking. In contrast, an upper limit policy differentiates access to resources on the overall relative importance of connections. The upper limit policy can also be readily implemented within minutes of a major event. The needs of important traffic can be met with approximately zero blocking very early in an event, allowing free access to resources in the early life-saving stages of a disaster.

## VIII. SUMMARY

This paper showed that prioritization of resources is indeed beneficial and that prioritized resource allocation can be implemented using simple algorithms. The paper presented a new upper limit policy methodology that optimized upper limit thresholds to provide preferred connection admission to high priority traffic classes based on a weighted sum of blocking metric which had not been used before. In related work, the upper limit policy was found to

TABLE 1 COMPLETE SHARING BLOCKING FOR OVERLOADED EXAMPLE SYSTEM

Blocking for Complete Sharing at Twice Normal Load C = 622 Mbs (OC-12), 8064 units of bandwidth (one unit = 64 kbs)	
Class 1: High Importance, Video Connection (1.28 Mbs), Weight = 0.3	$P_{B_1} = 0.9999$
Class 2: High Importance, Low Rate Video Connection (128 kbs), Weight = 0.3	$P_{B_2} = 0.5871$
Class 3: High Importance, Audio Connection (64 kbs, E-911), Weight = 0.3	$P_{B_3} = 0.3573$
Class 4: Medium Importance, Low Rate Video Connection (128 kbs), Weight = 0.08	$P_{B_4} = 0.5871$
Class 5: Low Importance, Voice Connection (64 kbs), Weight = 0.02	$P_{B_5} = 0.3573$
Sum of Weighted Blocking	$W_B = 0.637$

have many advantages, especially when trying to explicitly control blocking. Here a new optimization formulation for upper limit policies was derived from Kelly's approximation for asymptotically large networks [20]. The result was a simple linear program and a simple algorithm that finds the optimal upper limit policy for an arbitrarily large network with an arbitrarily large number of classes.

This paper was the first to compare the amount of capacity needed to implement resource policies and their sensitivity to load variations. The upper limit policy was demonstrated to use less than half of the resources of complete sharing to provide comparable weighted blocking during typical disaster overload conditions. The upper limit policy was also demonstrated to be less sensitive than complete partitioning to the large load variations that can occur in high priority traffic. When implemented along with the ticket server architecture in [5], public networks will be able to give preferred access to resources so that the important needs of society can be addressed when disasters or other special needs arise.

#### REFERENCES

- [1] Federal Emergency Management Agency, "Technology Applications by the Federal Emergency Management Agency in Response, Recovery, and Mitigation Operations," *Paper Presented to the 27th Joint Meeting of the U.S./Japanese Panel on Wind and Seismic Effects*, Tokyo/Osaka, Japan, May 16-27, 1995.
- [2] G. Philip and R. Hodge, "Disaster Area Architecture," *Proceedings of IEEE MILCOM*, 1995, pp. 833-837.
- [3] S. Adamson and S. Gordon, "Analysis of Two Trunk Congestion Relief Schemes," *Proceedings of IEEE MILCOM '93*, pp. 902-906, 1993.
- [4] Computer Science and Telecommunications Board (CSTB), National Research Council, *Computing and Communications in the Extreme: Research for Crisis Management and Other Applications*, National Academy Press, Washington, D.C., 1996.
- [5] C. Beard and V. Frost, "Dynamic Agent-Based Prioritized Connection Admission for Stressed Networks," 1999 IEEE International Conference on Communications, Vancouver, British Columbia, June 1999.
- [6] G. Foschini, B. Gopinath, and J. Hayes, "Optimum Allocation of Servers to Two Types of Competing Customers," *IEEE Transactions on Communications*, Vol. Com-29, No. 7, July 1981, pp. 1051-1055.
- [7] P. B. Key, "Optimal Control and Trunk Reservation in Loss Networks," *Probability in the Engineering and Informational Sciences*, 4:203-242, 1990.
- [8] J. S. Kaufman, "Blocking in a Shared Resource Environment," *IEEE Transactions on Communications*, Vol. COM-29, No. 10, October 1981, pp. 1474-1481.
- [9] K. W. Ross, *Multiservice Loss Models for Broadband Telecommunication Networks*, Springer-Verlag, London, 1995.
- [10] K. Ross and D. Tsang, "The Stochastic Knapsack Problem," *IEEE Transactions on Communications*, Vol. 37, No. 7, July 1989, pp.740-747.
- [11] H. Tijms, *Stochastic Models: An Algorithmic Approach*, John Wiley & Sons, 1994.
- [12] G. Foschini, B. Gopinath, and J. Hayes, "Optimum Allocation of Servers to Two Types of Competing Customers," *IEEE Transactions on Communications*, Vol. Com-29, No. 7, July 1981, pp. 1051-1055.
- [13] B. Kraimeche and M. Schwartz, "Circuit Access Control Strategies in Integrated Digital Networks," *IEEE INFOCOM '84*, San Francisco, CA, April 1984, pp. 230-235.
- [14] I. Gopal and T. Stern, "Optimal Call Blocking Policies in an Integrated Services Environment," Proceedings of the 17<sup>th</sup> Conference on Information Science and Systems, The Johns Hopkins University, Baltimore, MD, 1983, pp. 383-388.
- [15] S. Jordan and P. Varaiya, "Control of Multiple Service, Multiple Resource Communication Networks," *IEEE Transactions on Communications*, Vol. 42, No. 11, November 1994, pp. 2979-2988.
- [16] J. Hyman, A. Lazar, and G. Pacifici, "A Separation Principle Between Scheduling and Admission Control for Broadband Switching," *IEEE Journal on Selected Areas in Communications*, Vol. 11, No. 4, May 1993, pp. 605-616.
- [17] G. Choudhury, K. Leung, and W. Whitt, "An Algorithm to Compute Blocking Probabilities in Mult-Rate Multi-Class Multi-Resource Loss Models," *Advances in Applied Probability*, 27:1104-1143, 1995.
- [18] G. L. Choudhury, K. K. Leung, and W. Whitt, "Efficiently Providing Multiple Grades of Service with Protection Against Overloads in Shared Resources," *AT&T Technical Journal*, July/August 1995, pp. 50-63.
- [19] S. K. Biswas and B. Sengupta, "Call Admissibility for Multirate Traffic in Wireless ATM Networks,"
- [20] F. P. Kelly, "Blocking Probabilities in Large Circuit-Switched Networks," *Advances in Applied Probability*, 18: 473-505, 1986.
- [21] P. J. Hunt and F. P. Kelly, "On Critically Loaded Loss Networks," *Advances in Applied Probabilities*, 21: 831-841, 1989.
- [22] M. I. Reiman, "A Critically Loaded Multiclass Erlang Loss System," *Queueing Systems*, 9: 65-82, 1991.
- [23] M. I. Reiman, "Some Allocation Problems for Critically Loaded Loss Systems with Independent Links," *Performance Evaluation*, 13: 17-25, 1991.
- [24] A. A. Puhalskii and M. I. Reiman, "A Critically Loaded Multirate Link with Trunk Reservation," *Queueing Systems*, 28: 157-190, 1998.
- [25] N. G. Bean, R. J. Gibbens, and S. Zachary, "The Performance of Single Resource Loss Systems in Multiservice Networks," in *The Fundamental Role of Teletraffic in the Evolution of Telecommunications Networks*, eds. J. Labetoulle and J. W. Roberts, Amsterdam: Elsevier, 1994.
- [26] N. G. Bean, R. J. Gibbens, and S. Zachary, "Asymptotic Analysis of Single Resource Loss Systems in Heavy traffic, with Applications to Integrated Networks," *Advances in Applied Probability*, 27: 273-292, 1995.
- [27] P. J. Hunt and C. N. Laws, "Optimization via Trunk Reservation in Single Resource Loss Systems Under Heavy Traffic," *Annals of Applied Probability*, 7: 1058-1079, 1997.
- [28] Network Reliability Council, Focus Group 5, "Network Reliability -- The Path Forward: Telecommuting as a Back-Up In Emergencies", February 1996, Section 5.
- [29] FCC Operational Requirements Subcommittee, "Final Report, Presented To Public Safety Wireless Advisory Committee, National Telecommunications And Information Agency", May 29, 1996.
- [30] Telecommunications Service Priority, *Code of Federal Regulations*, Title 47, Chapter 1, Part 64, Appendix A.