Admission Control for Booking Ahead Shared Resources

Damon Wischik
University of Cambridge

Albert Greenberg
AT&T Labs–Research

Abstract— Calls that make large, persistent demands for network resources will be denied consistent service, unless the network employs adequate control mechanisms. Calls of this type include video conferences. Although overprovisioning network capacity would increase the likelihood of accepting these calls, it is a very expensive option to apply uniformly in a large network, especially as the calls require high bandwidth and low blocking probabilities. Such large calls typically require coordination of geographically distributed facilities and people at the end systems. So it is natural to book the network requirements ahead of their actual use. In this paper, we present a new, effective admission control algorithm for booking ahead network services. The admission control is based on a novel application of effective bandwidth theory to the time domain. Systematic and comprehensive simulation experiments provide understanding of how booking ahead effects call blocking and network utilization, considering call duration, number of links, bandwidth, routing, and the mix of bookahead versus immediate arrival traffic. Allowing some calls to book ahead radically reduces their chance of service denial, while allowing flexible and efficient sharing of network resources with normal calls that do not book ahead.

Keywords— Booking Ahead, Advance Reservation, Admission Control, Integrated Services Networks, Video Conferencing, Effective Bandwidths.

Lepidus. But small to greater matters must give way
Enobarbus. Not if the small come first
Antony and Cleopatra, Act II, Scene II

I. Introduction

In recent years, real-time multimedia network applications have received increasing attention. These applications often require a certain quality of service from the network, perhaps in terms of bounds on packet delay, jitter, or loss probability. In order to guarantee the quality of service, systems for reserving resources have been proposed. Under these systems, a call admission control algorithm keeps track of the amount of resources used by active calls, and will only admit new calls if there are adequate free resources.

This can lead to undesirable consequences. An online gaming connection, if it is set up first, could interfere with a video conference involving many people. If the organizers of a video conference cannot be reasonably certain that the resources will be there when needed, they will hesitate to set it up in the first place. A natural way for the network to guarantee resources in this way is to allow the call to be *booked ahead.*

The amount of bandwidth available is limited: this is presupposed by the need to reserve bandwidth in the first place. Booking ahead cannot create extra bandwidth; but if some calls are booked further in advance than others, those calls can be admitted in preference. In effect, calls can be given call-level priorities, based on when they book rather than when they start.

Booking ahead has several advantages over other call-level priority mechanisms. It is easily understandable to users, who will all have had experience of making reservations in their daily lives. It does not rely on overprovisioning of capacity. It moves control and responsibility away from the network and towards the user: instead of being restrained by a centrally imposed admission control policy, the user can book further in advance for more important calls. Knowing about future calls allows the network to better plan resource allocation.

Booking ahead has some disadvantages. It requires that the user plan in advance. However, the sort of calls that require large network resources often also involve a number of people and facilities, and would commonly be arranged in advance anyway. Another disadvantage is that other calls must be turned away to reserve space for the booked call, which in certain circumstances, can lead to reduced utilization: this impacts the cost to the network of booking ahead. Bookahead also requires that the network carry extra state information.

Our main contribution is a flexible, efficient call admission control algorithm that allows the calls and the network to benefit from the advance notice. The admission control algorithm is based on the theory of effective bandwidths. We also present the results of a systematic simulation study, which illustrate the particular benefits of booking ahead to large calls, and the impact of booking ahead on network utilization.

Outline

Section II outlines how we will measure the value of booking ahead for the user and for the network, and lists the requirements of a booking ahead admission control algorithm. In Section III we present our admission control algorithm, and explain its connection with effective bandwidths. We present simulation results in Section IV. Finally, in Section V we review previous work on booking ahead.

Damon Wischik is with the Statistical Laboratory, Mill Lane, Cambridge, UK. Email: D.J.Wischik@statslab.cam.ac.uk .

Albert Greenberg is at Florham Park, NJ 07932-0971, USA. Email: albert@research.att.com .

0-7803-4383-2/98/$10.00 © 1998 IEEE.

II. Background

A. Value of Booking Ahead

The reason for introducing booking ahead is so that certain users, typically those needing large amounts of resources such as for video conferences, can be sure in advance that resources will be available at the time they are needed. By booking ahead, these calls should have lower blocking probability: and we will measure the *effectiveness of booking ahead* in terms of this reduction.

We will see that in many cases, the reduction in blocking is a threshold effect: booking ahead a small time in advance has no effect, booking ahead a certain critical time in advance gives a dramatic reduction in blocking, and booking ahead further in advance has no further advantage. With sufficient notice, blocking hardly increases with the number of links occupied or with call bandwidth. This has bearing on the capacity planning of large networks. It shows that it is possible to carry high bandwidth or multicast calls without expensive overprovisioning.

Booking ahead would be of no use if everyone did it. We expect that there will always be spontaneous calls, because of the inconvenience of planning in advance, or because of limits imposed by the network. Bookahead calls should have lower blocking, so spontaneous calls will suffer increased blocking, resulting in some loss of traffic to the network. We will measure the *cost of booking ahead* in terms of the net impact on network utilization.

Suprisingly, it turns out that the net impact can be either an increase or a reduction in utilization. In other words, booking ahead might be a premium or a discount service. These two possibilities may be likened to two everyday uses of booking ahead. Some restaurants do not accept reservations, since it would leave tables reserved and empty while customers are waiting. If they did accept reservations, they might have to increase prices to compensate. On the other hand, booking an airline ticket in advance tends to be cheaper. The customer is rewarded for the trouble of planning in advance, which helps the airline in its planning.

B. Admission Control Algorithm Requirements

The key contribution of this paper is an admission control algorithm for booking ahead, based on the theory of effective bandwidths. This theory is a widely used and studied framework for statistical multiplexing at the packet level. We apply it instead at the call level (i.e., taking call durations into account). To our knowledge, effective bandwidths have not before been used in this way. Packet-level and call-level effective bandwidths share the same fundamental mathematics, but the two uses are quite independent.

In this section, we describe what is required of a bookahead admission control algorithm for an integrated services network. The aim of the algorithm is to integrate various different types of traffic while providing a suitable call-level quality of service guarantee, without excessive overheads. The meaning and implications of this aim are discussed in the following sections.

It is useful to bear in mind some different applications in the discussion of booking ahead. These include video conferences, audio conferences, broadcasts of planned events such as lectures or sports fixtures, video on demand, scripted presentations with distributed multimedia data, and the creation of private subnetworks. (The AT&T Accunet Bandwidth Manager is a system currently in use, for booking private subnetworks in advance). Non-bookahead calls that reserve resources should also be borne in mind, since they compete for the same resources. They include telephony and video broadcast.

There are some terms which are normally applied at the packet level, but which we apply at the call level. We have already mentioned effective bandwidths. Packet-level quality of service refers to cell loss probabilities; here it relates to call admission probability. Packet-level priority describes how the packets of one call are given priority at a queue; here it refers to which call books first.

B.1 Quality of Service

The purpose of booking ahead is to provide a more sophisticated system of call-level priority than the current model of 'turn up and pray'. This carries with it the notion of call-level quality of service: a measure of how likely it is that the user who has made a booking will get all the resources requested.

There are two broad classes of service guarantees: deterministic and statistical. A deterministic guarantee ensures that if a booking is accepted then the resources will be available. A statistical guarantee ensures only that there is a high probability that they will be available. We feel it is natural to provide a statistical guarantee, particularly in packet-switched networks where packet-level quality of service may only be statistically guaranteed. Statistical guarantees permit higher utilization. Our algorithm has a tunable parameter for call-level quality of service, which may be set globally or on a call-by-call basis. It may also be set to give deterministic guarantees.

With deterministic guarantees, the network cannot become overbooked. When the guarantees are statistical, it is necessary to have a policy for what to do when overbooking occurs. It turns out that the policy that is most natural is to degrade the service given to calls in reverse order to the order they were booked: most recently booked calls get degraded service first. Our quality of service parameter takes into account the chance of getting degraded service due to overbooking.

We write *preemption*, literally *being bought previously*, to refer to any sort of degraded service due to overbooking.

B.2 Duration

For many applications, including most of those that do not book ahead, it would be unreasonable to require that duration be specified in advance. For example, the length of a telephone call will depend on what each party says part way through, a football game could go into overtime, and a video-on-demand viewer might pause the video. But a basic assumption for any sort of booking ahead is that there is a model for the duration of any call that reserves bandwidth: if we do not know how long currently active calls are likely to last, we cannot accept bookings. Since our algorithm provides a statistical guarantee of service, it is natural to take as our model a distribution for the random duration of a call.

This information may be obtained in a number of ways. The simplest is when the user specifies the duration of the call in advance. Another approach would be for the network to assume a distribution, based on measurements of typical durations. Commercial networks have the data needed to do this; but it does not take into account what the user believes about the call duration.

In practice, some combination of the two is more likely. Users might tell the network what sort of call they are making: telephony, audio broadcast, etc. and the network might use typical durations for that type of call. Such decisions might be made explicitly, or implicitly based on a choice of pricing structure. Users might have default profiles, updated by their systems according to their individual measured durations, and passed on to the network.

It is worth noting that even advance bookings need not be for a strictly limited time. For example, a typical duration for a video conference might be expressed as *two hours plus a possible runover of a few minutes* which might be booked as *two hours plus an exponential duration of mean 10 minutes* according to whatever the users' profiles and network measurements suggest. In this way, preemption would be very unlikely for the first two hours, but become more likely the more the meeting overruns. Users should be able to cancel bookings, to try to extend bookings if the network has sufficient capacity, and to shorten bookings.

In what follows, we will assume that the distribution of the duration of each call is given. We will not be concerned with how it is obtained.

B.3 Bandwidth

Different applications will have different bandwidth requirements. A video-on-demand server might produce smoothed output, where the bit rates over different intervals are known in advance, in which case it makes sense to book precisely what is needed. For live video, the bandwidth might be highly variable, and there might be optional streams for higher quality.

In order to reserve bandwidth at all, we need to be able to measure it. While there is general agreement on the need to measure, there is not yet consensus on how. Candidate measures of bandwidth include *effective bandwidth* (applied at the packet level) and *measurement-based prediction of bandwidth* [1], in addition to the obvious *peak bandwidth.* We will not be concerned with how bandwidth is measured: our admission control algorithm applies generally, and to circuit-switched as well as packet-switched networks, though the guarantee of service is interpreted differently in each case.

B.4 Advance Notice

A small but important point is that the timescale of advance notice varies widely, from a few minutes for scripted presentations, up to weeks or months for booking private subnetworks. So, for example, a system based on video conferencing which clumps requests in half-hour intervals (e.g., [1]) might not extend well to other applications. Our algorithm gives a uniform mechanism for booking ahead at all timescales.

III. Admission Control

In this section we describe our model of bookahead admission control, and discuss briefly its assumptions and some simple extensions. We then show how the theory of effective bandwidths may be used to design the admission control algorithm.

A. Reservations and Arrivals

First, consider the simplest case, a single link. A call booking request consists of a bandwidth requirement, a starting time, a duration, and a quality of service parameter γ. Call admission control acts at the moment the booking is requested, and checks that the probability that the requested bandwidth will be available is high. Specifically,

$$\mathbb{P}(\text{get requested bandwidth for entire duration}) \geq 1 - e^{-\gamma}. \qquad (1)$$

If the booking is accepted, then at the booked start time the call itself arrives. It may happen that an arriving call finds insufficient available bandwidth. In this case, of the currently active calls the most recently booked is preempted.

In a network, a call requires bandwidth on each link along a route from its source to its destination. Assume that the route is selected at the time the booking is made. To accept the reservation, there must be sufficient available capacity at the start time on each link of the route. To make this determination, each link, acting independently of all other links, verifies that (1) holds, replacing the parameter $e^{-\gamma}$ with $1-(1-e^{-\gamma})^{\frac{1}{m}}$, where m is the number of links on the route. Thus, if as a heuristic we regard link preemption probabilities as independent, the call gets its requested bandwidth for its entire duration on each link of its route with probability $\geq 1-e^{-\gamma}$, as before. If an arriving call finds insufficient bandwidth on any link of its route, then that link independently applies call preemption, with priority given to the calls booked earliest. In our simulations, we found that the preemption probability of one of the links dominates, and it suffices to use the original criterion (1) on each link, with parameter $e^{-\gamma}$ rather than $1-(1-e^{-\gamma})^{\frac{1}{m}}$.

We have described stochastic guarantees of service. Of course, the system could offer a deterministic guarantee simply by setting $e^{-\gamma}=0$. This would involve some simple changes to the algorithms described in Section III-B.

When a booked call arrives and there is insufficient capacity, instead of preempting the call whose booking is most recent we could preempt the newly arriving call. However, this affects the preemption probabilities of calls whose bookings have already been accepted. As a result, quality of service guarantees (1) for these calls would have to be recalculated, which seems overly complex. A good admission control keeps the chance of preemption sufficiently small that the simple preemption policy given above should suffice, and the action taken on preemption should be of secondary importance.

We assume that the network computes a call's route when the booking is made. Alternatively, route computation could be carried out at any time up until the call arrives. Deferring route computation complicates overall admission control and signalling. Though these complications may win some improvements, the simulations of Section IV-G suggest that load-sensitive routing of calls at the time they are booked is an effective strategy.

The system would in practice have to cope with cancellations and calls which do not activate. The modifications to the algorithm are simple, but the implications for charging require further study.

B. Calculating Preemption Probability on a Single Link

To review, a booking is accepted if it is acceptable on each link of its route. To determine if the call is acceptable on a given link, we need to calculate the call's preemption probability if accepted. As earlier bookings take priority over later bookings, the decision whether or not to accept the new booking does not impact earlier decisions.

We will assume that the distribution for the duration of each call is known, and that call durations are independent random variables. Of course, if the duration is specified by the user, that value is used as a 'random' variable of fixed value.

It remains then to provide the details of preemption probability calculations on a single link. Suppose that at time u a new booking is requested, on a link having total capacity C. How do we estimate the associated probability of preemption in (1)?

First, consider the problem of estimating the preemption probability at a particular time t, at or after the booked start time. At time u, the link knows which of those calls that were previously booked might be active at time t: calls in progress at time u, and calls not yet in progress, but booked at some time $\leq u$ to start at some time $\leq t$. Indexing these calls arbitrarily, let a_i denote the start time, D_i the duration and c_i the required bandwidth of call i. Then,

$$\mathbb{P}(\text{new call preempted at time } t \mid \text{link state at time } u)$$
$$= \mathbb{P}\left(\sum_i c_i 1_{t \leq D_i + a_i} \geq C \mid D_i + a_i \geq u \;\forall i\right)$$

where the sum is over all calls potentially active at time t. By the Chernoff bound, this is

$$\leq \mathbb{E}\left(e^{s(\sum_i c_i 1_{t \leq D_i + a_i} - C)} \mid D_i + a_i \geq u \;\forall i\right) \quad \forall s > 0$$

and by independence of the D_i,

$$= e^{-sC} \prod_i \mathbb{E}(e^{s c_i 1_{t \leq D_i + a_i}} \mid D_i + a_i \geq u \;\forall i\,).$$

Require that this probability of preemption be

$$\leq e^{-\gamma}.$$

This is equivalent to

$$\sum_i c_i \beta_{D_i}(s c_i, t - a_i \mid u - a_i) \leq C^*(s) \quad \text{for some } s > 0, \tag{2}$$

where we define the effective bandwidth at time t, $\beta_D(s, t \,|\, u)$, and the effective capacity $C^*(s)$ by

$$\beta_D(s, t \mid u) = \frac{1}{s} \log\Big(1 + (e^s - 1)\mathbb{P}(D \geq t | D \geq u)\Big) \quad \text{and} \tag{3}$$

$$C^*(s) = C - \frac{\gamma}{s}. \tag{4}$$

Note that this calculation is conservative.

The new call is in danger of preemption only at times t coinciding with: its own start time, and the start times $a_i > t$ of previously booked calls. Summing the preemption probabilities at these times gives an upper bound on the new call's preemption probability. However, we know from extensive simulations that the largest preemption probability among these times provides a sharp estimate of the overall preemption probability.

Thus, our link admission control is defined as follows. The link keeps track of the accepted calls: active calls, and calls booked but not yet active. When a new booking is submitted, requesting a certain quality of service specified by γ, the link checks if there is sufficient capacity, by numerically solving (2) for each time t where the call is in danger of preemption, as just discussed. If the booking is admissible at each of those times, it is accepted on the link.

Figure 1 illustrates an admission control decision. The effective bandwidths (for a particular value of s) of the various calls are superimposed, and the effective capacity is shown. The effective bandwidths give a measure of how likely it is that a call will be active at a certain time in the future, and they incorporate everything we know about the call. In this example, the total effective bandwidth including the new call is less than the effective capacity, so we accept it.

Though the effective bandwidth profile changes continuously, calculating call preemption probabilities is not burdensome. Inequality (2) holds if the preemption probability estimate

$$\inf_{s>0}\left(\sum_i s c_i \beta_{D_i}(s c_i, t - a_i \mid u - a_i) - sC\right) \tag{5}$$

is sufficiently small. Let s^* denote the value of s that minimizes (5). It is not hard to show that s^* lies between the maximum and minimum values of the quantities

$$\log\Big(\frac{1 - \mathbb{P}(D_i \geq t \mid D_i \geq u)}{\mathbb{P}(D_i \geq t \mid D_i \geq u)}\Big) - \log\Big(\frac{B}{C} - 1\Big)$$

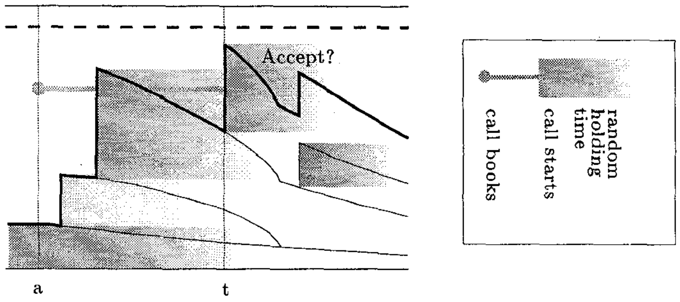

Fig. 1. A new booking is submitted to a link at time a, for a call to start at time t. Do we accept it? Time increases along the horizontal axis, and used bandwidth along the vertical axis. A call is represented by a rectangle, and the bandwidth required by its height. As call durations are random, at time a there is uncertainty as to when calls depart, represented by the fading of rectangles as the associated calls age. The line that rises from the leading edge of a rectangle and then decreases with time represents the call's *effective bandwidth*, defined in this section. Summing the effective bandwidths gives the bold line. The horizontal dashed line shows the link's *effective capacity*, also defined in this section. In this example, the total effective bandwidth is less than the effective capacity, so the booking is accepted.

where B is the total bandwidth in use on the link. In practice, searching for s^* within this range is straightforward. By definition of s^*, if (5) is sufficiently small for some s, then it is also for $s = s^*$. Thus we can, for example, continue to use the same s for successive booking requests until a booking is rejected, at which point we recalculate s^* from the current link state.

C. Relation to Effective Bandwidth

Effective bandwidths are widely studied [2], [3] and used at the packet level. They provide an additive measure of bandwidth, between the mean and peak rates, which incorporates information about burstiness and multiplexing gain. We apply the theory here instead at the call level. The call-level effective bandwidth $\beta(s, t \mid u)$ defined in (4) is an additive measure of bandwidth, less than or equal to the requested bandwidth, indicating how likely it is that the call is still active. But the two senses should not be confused. For example, t refers to real time, but the time parameter appearing in [2] refers to characteristic timescales.

Our results benefit from another characteristic of effective bandwidths. While the acceptance region given by (2) is conservative, it is also asymptotically exact, in the limiting regime where the number of calls of the different classes and the capacity tend to infinity in proportion. In this limiting sense, our algorithm is optimal. For details of the theory behind this sort of limiting regime, see for example [4].

If we are also using effective bandwidths to characterize the source at a packet level, the calculation is slightly altered. There are many forms of effective bandwidth, and we look at a bufferless, discrete time model by way of illustration. Assume that, for a call starting at time a with duration D, the number of packets generated at time t is $X(t)1_{a+D\geq t}$ where $X(t)$ is a stationary process independent of D. Then, for a queue that can serve C packets per unit time,

$$\begin{aligned}&\mathbb{P}(\text{overflow at time } t \mid \text{link state at time } u)\\&= \mathbb{P}\Big(\sum_i X_i(t)1_{t\leq D_i+a_i} \geq C \mid D_i + a_i \geq u \;\forall i\Big)\end{aligned}$$

and the calculation proceeds as before. Let $\alpha_X(s) = (1/s)\log \mathbb{E}e^{sX(0)}$ be the conventional effective bandwidth for a source X. Our effective bandwidth for this source truncated to have duration D is now

$$\beta_{D,X}(s, t \mid u) = \frac{1}{s}\log\Big(1 + (e^{s\alpha_X(s)} - 1)\,\mathbb{P}(D \geq t | D \geq u)\Big).$$

This concludes the discussion of our call admission control algorithm. In the next section, we evaluate its performance. Though the algorithm has an analytic basis, we have no analytic results for its performance. Instead, to understand its effect on blocking and utilization, we have conducted extensive simulation experiments.

IV. Simulation Results

In this section we use simulation to explore the implications of booking ahead. As described in Section II-A, we will measure the reduction in blocking for a class of calls which books ahead, and the total utilization of the network. We always assume a class of calls which does not book ahead, and suffers increased blocking as a consequence.

In Section IV-A we look at an example of how booking ahead reduces blocking in a network. The notable feature of this example is the phenomenon of *critical bookahead times.* We simplify the model in various ways in Sections IV-B, IV-C and IV-D to explore and explain this phenomenon. Section IV-E illustrates how the reduction in blocking achieved by booking ahead is most dramatic for large calls.

In Section IV-F we look at the impact of booking ahead on utilization, and see that the impact of booking ahead may be either to increase or decrease total utilization. We compare utilization under deterministically and statistically guaranteed service.

Finally, we look at the importance of quality-of-service based routing in Section IV-G.

Throughout the simulations, we observed only a handful of cases of preemption, which was for simplicity simulated as dropping the preempted call entirely. The quality of service parameter was set to 1% chance of preemption throughout, and is clearly conservative.

The traffic models we report are chosen to illustrate the behaviour of our admission control algorithm. Our results are typical of a large number of simulation scenarios; but the algorithm applies equally whatever the nature of the traffic: periodic, statistically concentrated at half-hour intervals, or anything else.

A. A Network Example

Consider the network of Figure 2, modelled after the topology and load of a fragment of a large AT&T integrated services network [5], which provides a type of bookahead service.

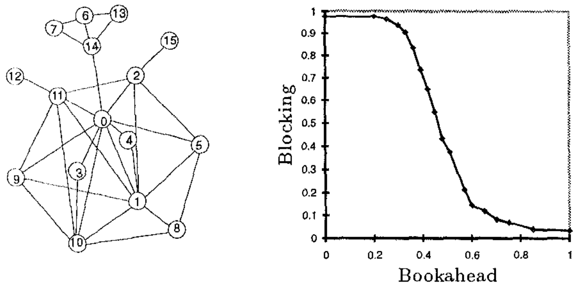

Fig. 2. An example network, modelled after the topology and load of a fragment of a large AT&T network. The bottleneck link between nodes 0 and 14 is highly congested and runs at about 95% utilization. Calls along the route 7-14-0-10 request 30% of that bandwidth. If they do not book ahead, they have little chance of being accepted. As they book further in advance, their blocking reduces dramatically. This behaviour is seen in a wide range of network scenarios, whether or not the bookahead call uses a bottleneck link.

There are 190 different classes of calls which do not book ahead, and there is one class which does. Calls from each class arrive as independent Poisson processes with different rates. (Paxson and Floyd [6] have found that Poisson arrivals is a good model for user-initiated calls, and it is reasonable to assume that the real-time calls which require reservation are user-initiated). Non-bookahead calls have exponential holding times, bookahead calls have holding times uniformly distributed over a short interval. Each of the non-bookahead call classes has associated with it a set of possible routes. When a call arrives, the network computes which of these routes it might be accepted on. The call takes the route with the least number of hops, breaking ties randomly.

Remarkably, many of the features seen in simulation scenarios for this network also appear in the much simpler single-link case. Blocking tends to decrease in jumps: there is a *critical times* at which blocking decreases suddenly. In Sections IV-B, IV-C and IV-D we simplify the call model in various ways to explain this phenomenon.

B. Critical Bookahead Times

In this section, we present a very simple example showing critical bookahead times. In Sections IV-D and IV-C we will see that these simplifications are not essential.

Consider a single link accepting two classes of call: the Book Ahead (BA) class, and the Immediate Request (IR) class which does not book ahead. IR calls arrive as a Poisson process and have exponential holding times. The arrival times and holding times of the BA calls are uniformly distributed in such a way as to ensure that no more than one BA call is active at any time. This is a controlled way of modelling infrequent BA traffic.

Figure 3 shows that BA blocking decreases as BA calls book further in advance, with a dramatic decrease in blocking at a *critical bookahead time.* Additional notice gives no benefit. In other words, with sufficient advance notice, a call gets absolute priority. This means that it is very important to know what the critical bookahead time is: if the user does not book ahead far enough, nothing is achieved at all.

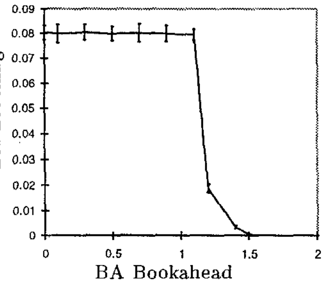

Fig. 3. Typical results for blocking of BA calls as bookahead time increases: blocking drops rapidly at a critical bookahead time. Error bars show 99% confidence intervals. The link has capacity 6; BA calls form an on/off source, with bandwidth 4, holding times are Uniform(.8, 1.2) and interarrival times are Uniform(3, 5); IR calls have bandwidth 1, holding times are Exp(2.7), and they arrive as a Poisson process of rate 2.7.

This may be understood as follows. BA calls are competing solely with IR calls, since the arrival process and holding times are arranged so that only one BA call can be active at any time. When presented with a booking request for a call starting time t ahead, the network calculates the probability p that sufficient IR calls will have drained within t. If p is below a threshold, the booking is accepted. Equivalently, if t is above a threshold t^* (depending on the current state of the link) the booking is accepted. When the IR calls have exponential holding times, then by the memoryless property of the exponential distribution t^* depends only on the number of current number of active IR calls, n. This will typically be at some stationary value. Furthermore, for the exponential distribution we heuristically expect that $t^* \sim \log(n)$ (by the nature of exponential decay, if there are twice as many currently active calls, we need only wait a single extra time unit for them to clear). So t^* should be fairly insensitive. The typical value of t^* is the critical bookahead time.

Critical times will occur in any large stable network. By *stable*, we mean that the distributions of the number of calls in the system, how long they have been active, how many calls are booked ahead, and all other aspects of network state, are stationary. By *large*, we mean that the system is large enough to have fluid-like behaviour, concentrated at the mean behaviour. If these conditions are satisfied, then the link state that any booking sees will be fairly constant, concentrated at the most likely state, and deviations will be rare. This is the *large deviations limit* of the network. The most likely state has a critical time t^*. In other words, if the link is in the most likely state and a booking arrives with advance notice less than t^* it will not be accepted, while if the advance notice is greater than t^* it will be. Since the network is large and stable, the critical time seen by any bookahead call will be very close to t^*.

C. Sensitivity to IR characteristics

The results of the previous section do not rely on the choice of a particular model for IR traffic. In this section, we see that there is still a critical bookahead time when IR arrival

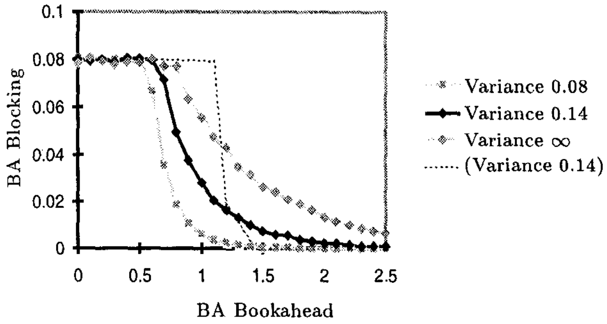

Fig. 4. The dotted line shows blocking when IR holding times are exponential, with the same parameters in Figure 3; and the solid lines show blocking when they are Pareto of the same mean as the exponential case but different variances.

rates and mean holding times change, and even when IR durations have a heavy-tailed distribution. We will start with the model of the previous section, and alter IR parameters.

Consider first the effect of changing IR arrival rate while leaving IR holding time constant. When the load (ratio of mean holding time to mean interarrival time) is 1 as in Figure 3, BA blocking is 0.08 when there is no bookahead, and it reduces to 0 around the critical bookahead time of 1.2. As the load is changed, BA blocking with no bookahead ranges from 0.19 when load is 1.48, to 0.04 when load is 0.74. But the critical bookahead time does not change at all.

Consider now changing IR arrival rate and IR holding time in proportion, keeping load fixed at 1. Now, by contrast, BA blocking with no bookahead is not affected, but the critical bookahead time ranges from 1.7 when arrival rate is 2, to 0.9 when arrival rate is 4.

We see that the critical bookahead time depends mostly on IR holding time. This may be understood from the discussion in Section IV-B. The main influence on the critical bookahead time is the holding times of the calls in the system when a booking request arrives; the number of calls in the system has little impact. The other quantity of interest, the blocking probability when there is no bookahead, may be found as a function of load and bandwidth by conventional techniques such as Erlang's formula [7]. Together, the critical bookahead time and the blocking probability when there is no bookahead give a very full description of how booking ahead affects blocking.

Figure 4 depicts results for the case where IR holding times are drawn from the Pareto distribution, with mean holding time kept fixed and variance changing. The Pareto(a, α) distribution, with density $f(x) = (\frac{x}{a})^{\alpha}$ for $x \geq a$, is a standard example of a heavy-tailed distribution. Heavy-tailed distributions have been suggested as good models for call holding times by Duffy *et al.* [8] (though they suggest that it may be data calls which can last for a long time, rather than interactive user-driven calls—and it is the latter that would be more likely to reserve resources).

The dotted line gives as reference the blocking when holding times are exponential with the same mean. There is still a critical bookahead time before which booking ahead gives no benefit. Blocking tails off more gradually for Pareto holding times than for exponential holding times; and for holding times of greater variance. Interestingly, the critical bookahead time is smaller when holding times are Pareto. In that sense, Pareto holding times are easier to cope with.

Again, the discussion of Section IV-B suggests why the tail off in blocking is more gradual when holding times are Pareto. The critical bookahead time t^* depends on the number of IR calls active and also now on how long they have been active, and so we expect greater variability in t^* and therefore a less dramatic decrease in blocking. But blocking decreases sharply at first, so there is still some threshold behaviour.

D. Concurrent BA calls

Sections IV-B and IV-C assumed an artificial model for BA arrivals and holding times, which guaranteed that no more than one BA call was active at any time, and simplified the discussion. In this section we look at two more natural models of BA call arrivals which lead to concurrent BA calls. Now, blocking cannot be totally eliminated, as the BA calls may block each other. However, the same general principles apply.

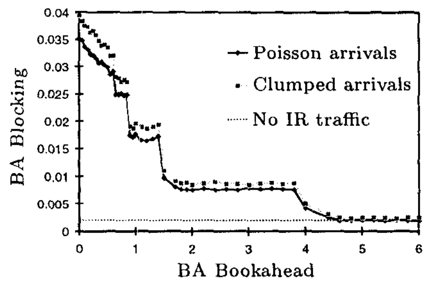

Fig. 5. Blocking for BA calls as bookahead time increases. BA calls arrive (a) as a Poisson process of rate .9 and (b) as a clumped process in which a Poisson number of calls (of mean .45) arrive every .5 time units. The link has capacity 5.5 and all calls have bandwidth 1. BA holding times are Uniform(.8, 1.2); IR calls arrive as a Poisson process of rate 2.9 and holding times are Exp(2.7). Also shown is the blocking when there is no IR traffic, which is nearly identical in the two cases.

Figure 5 shows how BA blocking decreases as bookahead time increases for two different BA arrival processes. In the first model, BA calls arise as a Poisson process. In the second, a random number of calls arises every .5 time units; this is to illustrate what happens when many bookings are made for the same start time—for example, video teleconferences are more likely to start on the hour than at randomly distributed points in time. There is very little difference between the two cases.

In both cases, if the BA calls book far enough in advance (5 time units) then all blocking due to IR calls is eliminated.

A maximum of 5 BA calls can be in the system at any one time; and the blocking curves shows threshold behaviour at 5 bookahead times, the critical bookahead times for getting access to those 5 units of capacity. In the same way that booking a larger proportion of capacity requires a longer bookahead time for the on/off source, so here booking successive units of capacity requires longer bookahead times. The thresholds will be smaller and more numerous for a link of greater capacity. But they will still be apparent, particularly when the load is sufficiently high that there is little spare capacity.

The same behaviour is seen when IR calls have Pareto holding times, although the critical bookahead times are less well defined, as Figure 4 would suggest.

E. Large Calls

By large calls, we mean calls which occupy many links or which have high bandwidth. It is hard to provide adequate service to large calls when bandwidth is limited—indeed this was a major motivation for our study of booking ahead. In this section we see that booking ahead a little bit further in advance is sufficient to give low blocking for very much larger calls.

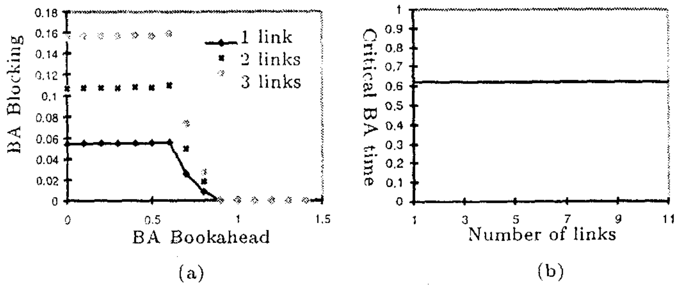

Fig. 6. Blocking for multilink BA calls reduces as bookahead time increasės, and the critical bookahead time does not increase with number of links. The link has capacity 15; BA calls form an on/off source of bandwidth 3, with interarrival times Uniform(3.6, 4.8) and holding times Uniform(.8, 1.2); IR calls have bandwidth 1, arrive as a Poisson process of rate 8 and have holding times Exp(1). BA calls require a variable number of links, each of which has its own IR traffic.

Figure 6 shows what happens when the BA call requires more than one link, where each link carries an independent stream of IR traffic. As the number of links required by the BA calls increases, the critical bookahead time does not change. This may be understood as follows. If there is minimal blocking b on a single link, there will be minimal blocking $1-(1-b)^n \approx nb$ for a fixed number n of links. At the critical bookahead time, $b \to 0$ so $nb \to 0$. In contrast, with no booking ahead, $b > 0$ and so there is significantly more blocking on n links.

The point to note is that by booking far enough in advance, all blocking due to IR calls can be eliminated. Blocking on BA calls is as if there were no IR calls at all: this is what helps multilink calls to be accepted. This is seen in all the examples we present. Short of increasing capacity, there is nothing more that can be done to reduce BA blocking.

There is a similar story for the effect of increasing bandwidth. The critical bookahead time increases slightly as BA bandwidth increases, but it varies little over most of the range.

These results suggest that booking ahead is very well suited to multimedia applications, which often require large bandwidths and multiple links. The topology of the links is not important, and so these results apply to multicasts. By booking ahead, blocking can be significantly reduced on large calls; and the necessary bookahead time does not grow as fast as the size of the call. On the other hand, booking ahead does not give fine control over priorities; and a small bookahead time gives no benefit.

F. Impact on Utilization

Whether it is profitable for the network to allow booking ahead, or what sort of charges might be levied, depends on the answer to the question: What is the impact of booking ahead on network utilization? It turns out that utilization may either increase or decrease as a result of allowing one call class to book ahead. The effect is subtle and hard to predict in a general network with multiple call classes; so in order to illustrate the range of effects, we will restrict attention to a single link with two classes of traffic, BA and IR, as in Section IV-B.

It is easy to imagine extreme cases which lead to a net increase in utilization. Suppose BA calls require 90% of the capacity, and the typical IR traffic takes up 11%. Then if the BA call books ahead far enough, 1% of the IR traffic will be blocked to make room, and total utilization will be near 100%. On the other hand, if each IR call is very large and takes up 91% of the capacity, and a BA call requests 10%, then accepting the BA call would give far lower utilization than the typical figure of 91%. This simplistic discussion does not take account of the fact that calls arrive and depart randomly, which would tend to reduce utilization. But the same effects can be seen in the following examples.

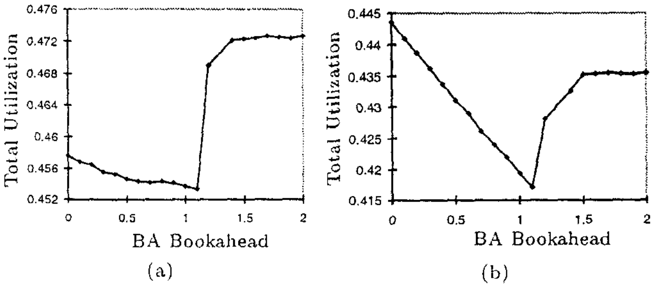

Fig. 7. Total utilization may either increase or decrease with booking ahead. Parameters are as for Figure 3; but for (a), BA holding times have been increased to Uniform(1.5, 2.5); amd for (c), IR arrival rate has been increased to Exp(6).

Figure 7 illustrates the range of effects that booking ahead can have on utilization. To understand why these different effects occur, it is useful to break down total utilization into two parts: that due to BA calls, and that due to IR calls. Utilization due to BA calls increases as BA bookahead increases, while utilization due to IR calls decreases. These two effects—the reduction in IR utilization and the increase in BA utilization—combine to give the net change in utilization. Whether there is a net increase or a net decrease will depend on the relative strengths of the two effects. When there is much potential BA traffic, a net increase in utilization is likely; when there is very little, a net decrease is more likely. In Figure 7 (a), BA traffic makes up 67% of all traffic (when it does not book ahead) and there is a net increase in utilization; in (b) it makes up 23% and there is a net decrease in utilization.

Another feature of the impact of booking ahead on utilization demonstrated by these results, is that it is a bad idea to book ahead less than the critical time. Not only does it fail to reduce blocking, but it also leads to lower utilization. The network might therefore refuse to accept such bookings. This suggests, for example, that booking ahead is an inappropriate mechanism for renegotiating bandwidths at very short timescales.

An alternative to our multiplexing admission control would be to partition the capacity into separate parts for BA and IR calls, which would give deterministic guarantees of service. In general it is unclear how bandwidth should be partitioned, but in this simple case where only one BA call can be active at a time it is natural to reserve the BA bandwidth for BA traffic and leave the rest to IR traffic. This gives poorer utilization: in these examples, around 1% less than the lowest utilization observed with multiplexing.

G. Routing and Rerouting

In this experiment, we investigate the importance of routing. When there is a choice of routes, there are two natural strategies, which we will call ROUTE and REPACK. Under ROUTE, when a BA booking arrives, the preemption probabilities on each of the possible routes is calculated, and the booking is assigned the route with the lowest preemption probability. Under REPACK, BA bookings are initially routed in the same way as ROUTE, but they may be rerouted before the call is due to start if more traffic arises on the initial route. In fact, since REPACK requires sophisticated techniques to decide when bookings should be rerouted, we will consider instead the system POOL. This combines all links into one, and so gives a bound on how good any repacking scheme can be. These strategies will be compared with RANDOM, which assigns bookings to routes randomly. This is to mimic the effect of routing without any quality of service information.

Figure 8 (a) shows blocking on BA calls for the three strategies RANDOM, ROUTE and POOL, in the simplest case with routing choice: two links, each with its own dedicated stream of IR traffic, and BA calls able to be routed over either of the links. There is a dramatic reduction in blocking obtained by ROUTE, and virtually no further reduction with POOL. In other words, quality of service based routing gives dramatic improvements, but there appears to be no benefit in repacking calls.

Figure 8 (b) shows that not only does ROUTE give much lower blocking than does RANDOM, it also gives higher utilization. Indeed, for a fair comparison, the capacity in the RANDOM case should be increased to reduce blocking to a comparable level to the ROUTE case: this would show the utilization in the ROUTE case even more favourably.

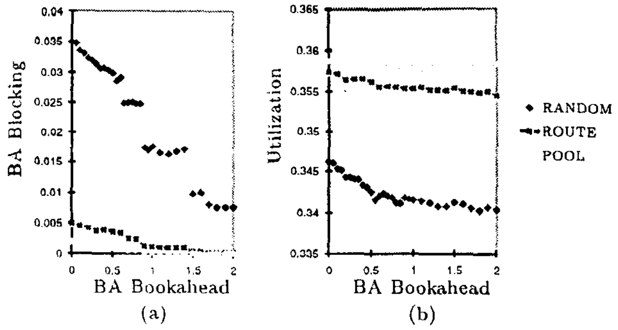

Fig. 8. The reduction in blocking (a) and the increase in utilization (b) due to best-choice routing in a two-link network. Parameters are as for the Poisson BA traffic, in Figure 5. The three curves illustrate three different routing strategies, described in the text.

These results still hold in networks, such as the network of Figure 2. In order to give scope for routing, we choose as the BA class calls between nodes 9 and 10, which may be routed through either 0 or 11. Now, ROUTE still performs better than RANDOM, and the improvement increases as BA bookahead increases: with no booking ahead, RANDOM gives 95% blocking while ROUTE gives 80%, and booking ahead 1 time unit, RANDOM gives 80% blocking while ROUTE gives 5%.

This is an example of a general phenomenon: only a small amount of routing flexibility is needed to achieve resource pooling. We believe that in general, issues of routing are independent of booking ahead, in that qualitative results carry through.

V. Previous Work

In this section we review three different approaches that have been taken to bookahead admission control, and place ours into context. The principal feature of our admission control algorithm is that it integrates calls of random duration with bookahead calls. The most useful classification of previous approaches to booking ahead is by how they deal with calls of random duration.

Degermark, Köhler, Pink and Schelén [1] sidestep the problem by assuming that all calls, whether booked in advance or immediate, declare their durations in advance. Ferrari, Gupta and Ventre [9] describe a partitioning system, which is effectively the same as treating all calls of random duration as though of infinite duration. The approach which is closest to ours is that of Greenberg, Srikant and Whitt [10]. They probabilistically model the durations of calls of random duration to calculate preemption probabilities. But their admission control calculation has the drawbacks that all random durations are taken to have the same distribution, and no account is taken of how long a call has been in the system. We have extended their framework, and eliminated these drawbacks with a novel admission control algorithm.

There are also differences in the measurement of bandwidth. Degermark *et al.* use measurement-based predictive service, which estimates bandwidth requirement in packet-switched networks based on past usage. Ferrari, Gupta and Ventre, and Greenberg, Srikant and Whitt look at an essentially circuit-switched model with the natural measure of bandwidth. We point out that booking ahead is basically independent of how bandwidth is measured, and suggest a variety of different ways of measuring bandwidth including effective bandwidths.

Various authors have looked at how existing resource reservation protocols might be extended to support booking ahead. Degermark *et al.* look at the proposed Internet protocol RSVP, Reinhardt [11] looks at another proposed internet protocol ST2, and Ferrari *et al.* look at the Tenet suite. Additional discussion of general issues relating to booking ahead may be found in [12], [13], [14], [15]. While conclusions as to the suitability of the various protocols differ, we note that our admission control algorithm does not require any additional control traffic beyond that which the above researchers have found necessary.

VI. Conclusion

We have presented an admission control algorithm for booking ahead. The algorithm provides statistical multiplexing at the call level, based on a conservative calculation of the preemption probability of a call. The theoretical basis is the theory of Effective Bandwidths, which is commonly applied to the packet level, but which we now see is also successful at the call level, taking call durations into account. Our algorithm can be used in conjunction with a variety of ways of measuring bandwidth.

We have seen that booking ahead can reduce blocking dramatically, if done sufficiently far in advance. Not booking far enough in advance gives little benefit to the user, and is detrimental to the network's utilization and stability. Incentives and control mechanisms are therefore important. Network controls might take the form of restricting bookahead time, or reserving capacity for non-bookahead calls. The results we have presented can help guide the selection of appropriate bookahead times and policies.

Booking ahead can lead to either an increase or a decrease in utilization, depending on the traffic mix. Pricing for booking ahead would need to reflect which of these occurs. In addition, since our admission control algorithm is probabilistic, it relies on modelling the call length distribution; and so the pricing framework could also have the functions of conveying duration information to the network, and of encouraging the user to be honest. Further work is required to understand how these various functions may be combined.

Designing a network which can cope with large calls poses special problems, which are alleviated by allowing calls to book ahead. Booking ahead is a natural way to increase the priority of a call. It does not require overprovisioning of capacity; and it is flexible in that policy decisions about how far ahead to book are left to the user. If booked sufficiently far in advance, calls using many links or high bandwidth need not experience excessive blocking. The advance notice needed does not increase with the length of a call's route, and only increases very slowly with the call's bandwidth.

Booking ahead is a valuable addition to the repertory of network control techniques. We have shown how admission control for booking ahead might be implemented, and simulation results illustrate the benefits that booking ahead can bring.

References

[1] Mikael Degermark, Torsten Köhler, Stephen Pink, and Olov Schelén, "Advance reservations for predictive service," in *Proc. NOSSDAV'95*, 1995.

[2] Frank Kelly, "Notes on effective bandwidths," in *Stochastic Networks: Theory and Applications*, F. P. Kelly, S. Zachary, and I. Ziedins, Eds., Royal Statistical Society lecture note series, chapter 8. Oxford, 1996.

[3] R. Guérin, H. Ahmadi, and M. Naghshineh, "Equivalent capacity and its application to bandwidth allocation in high-speed networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 7, pp. 968–981, September 1991.

[4] Amir Dembo and Ofer Zeitouni, *Large Deviations Techniques and Applications*, Jones and Bartlett, 1993.

[5] A. Greenberg and R. Srikant, "Computational techniques for accurate performance evaluation in multirate, multihop communication networks," *IEEE/ACM Transactions on Networking*, vol. 5, no. 2, pp. 266–277, apr 1997.

[6] Vern Paxson and Sally Floyd, "Wide-area traffic: The failure of Poisson modelling," in *Proc. SIGCOMM'94*. ACM Special Interest Group on Data Communications, 1994.

[7] F.P. Kelly, "Loss networks," *The Annals of Applied Probability*, vol. 1, no. 3, pp. 319–378, 1991, Special invited paper.

[8] Diane E. Duffy, Allen A. McIntosh, Mark Rosenstein, and Walter Willinger, "Statistical analysis of CCSN/SS7 traffic data from working CCS subnetworks," *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 3, pp. 544–551, 1994.

[9] Domenico Ferrari, Amit Gupta, and Giorgio Ventre, "Distributed advance reservation of real-time connections," in *Proc. NOSSDAV'95*, 1995.

[10] A. Greenberg, R. Srikant, and W. Whitt, "Resource sharing for book-ahead and instantaneous-request calls," in *Proc. 15th International Teletraffic Congress*, 1997, pp. 539–548.

[11] Wilko Reinhardt, "Advance reservation of network resources for multimedia applications," in *Proc. International workshop on advanced teleservices and high speed communication architectures (IWACA) '94*, 1994.

[12] Wilko Reinhardt, "Advance resource reservation and its impact on reservation protocols," in *Proc. Broadband Islands '95, Dublin, Ireland*, sep 1995.

[13] Lars C. Wolf, Luca Delgrossi, Ralf Steinmentz, Sibylle Schaller, and Hartmut Wittig, "Issues of reserving resources in advance," in *Proc. NOSSDAV'95*, 1995.

[14] Domenico Ferrari and Amit Gupta, "Admission control for advance reserved real-time connections," Tech. Rep., International Computer Science Institute, Berkeley, 1995.

[15] Amit Gupta, "Advance reservations in real-time commication services," Technical Report TR-96-0228, Sun Microsystems Laboratory, Mountain View CA 94043, May 1996.