Networks With Advance Reservations: The Routing Perspective

Roch A. Guérin and *Ariel Orda*

Abstract— **This paper provides an initial look at how support for advance reservations affects the complexity of the path selection process in networks. Advance reservations are likely to become increasingly important as networks and distributed applications become functionally richer, and there have been a number of previous works and investigations that explored various related aspects. However, the impact of advance reservations on path selection is a topic that has been left largely untouched. This paper investigates several service models for advance reservations, which range from the traditional basic model of reserving a given amount of bandwidth for some time in the future, to more sophisticated models aimed at increasing the flexibility of services available through advance reservations. The focus is primarily on the issue of computational complexity when supporting advance reservations, and in that context, we derive a number of algorithms and/or intractability results for the various models we consider.**

Keywords— **Routing, Networks, Advance Reservations, Bandwidth, Delay, Preemption.**

I. Introduction

A. Motivations

As networks capabilities increase, their usage is also expanding. At the same time, the wide range of requirements of the many applications using them calls for new mechanisms to control the allocation of network resources. For example, resource reservation as provided by the RSVP protocol [3], and service differentiation as embodied in the Integrated Services and Differentiated Services specifications [27], [24], [11], [12], have become important enablers for distributed applications.

However, while much attention has been devoted to resource reservation and allocation, the same does not apply to the timing of such requests. In particular, the prevailing assumption has been that requests are "immediate", i.e., made at the same time as when the network resources are needed. This is a useful base model, but it ignores the possibility, present in many other resource allocation situations, that resources might be requested in *advance* of when they are needed. This can be a useful service, not only for applications, which can then be sure that the resources they need will be available, but also for the network, as it enables better planning and more flexible management of resources.

R. Guérin (guerin@ee.upenn.edu) is with the Dept. Elec. Eng., U. Pennsylvania, 200 South 33rd Street, Philadelphia, PA 19104. Part of this work was done while at the IBM T.J. Watson Research Center.

A. Orda (ariel@ee.technion.ac.il) is with the Dept. Elec. Eng., Technion - I.I.T., Haifa, 32000 - Israel. Part of this work was done while visiting the IBM T.J. Watson Research Center.

Providing support for advance reservations in networks requires enhancements in three major areas:

1. protocol and signalling capabilities to allow users to express advance reservation requests;
2. extensions to resource management and call admission to handle time shifted resource requests;
3. routing algorithms to compute paths based on both spatial and temporal request characteristics.

Most efforts in the area of advance reservation have dealt so far with problems from the first and second areas, with little work having been done on the third, despite its potential importance. This third area is the focus of this paper. Before we proceed with a brief outline of the problems tackled in this paper, we rapidly review previous works and highlight their relevance to issues we investigate.

B. Previous Work

Some of the earliest work on advance reservation was done in the context of video-conferencing and satellite systems. Early video-conferencing systems involved high bandwidth signals between (fixed) video-conferencing sites (studios), and advance reservations were needed to ensure that adequate bandwidth was available. Similarly, early satellite systems offered the option to rent the use of transponders for specific amounts of time, which also required support for advance reservation. In those early systems, the bulk of the work [18], [19], [14], [13], [25] focused on traffic modeling and call admission procedures, to properly size such facilities. Some recent studies [5], [21], [26] have extended these early works from the circuit-switched environment they assumed, to that of modern integrated packet switching networks.

Most other works dealing with advance reservation have focused on extensions to signalling protocols, or formulated frameworks (including signalling and resource management capabilities) to support advance reservations. For example, [16] discusses extensions to the ST-II protocol [6] to allow advance reservations. References [23], [22] present a similar discussion on implementing an advance reservation system on top of RSVP [3]. In [7], the authors describe a proposal to enable advance reservations on top of the Tenet Suite 2 [2], while [17] outlines a similar system using either the ST-II or RSVP protocols. The approach of [20] proposes an agent-based infrastructure for advance reservation that avoids a complete upgrade to existing routers.

C. Focus of This Paper

The focus of this paper is, as stated earlier, the impact of advance reservations on the path selection process. Specifically, knowledge of *future* loads on network links is now another important parameter to consider when attempting to select a path for a new request. This is highlighted in [26, Section IV-G], which shows the improvement in blocking (request denial) achieved when using reservation-sensitive routing. However, [26] deals primarily with the call admission process, and discusses only briefly the impact on path selection. In contrast, this paper provides a comprehensive investigation of the relation between routing and advance reservations. The investigation focuses on describing possible extensions to path selection algorithms in order to make them advance-reservation aware, and on evaluating the added computational complexity introduced by these extensions. In addition, the paper also explores a number of variations on reservation models aimed at enhancing the range of services that can be supported. In that context, the focus is again on the impact on path selection and the associated computational complexity.

The previous paragraph outlined what the paper is about, but it is equally important to articulate what it is not. Specifically, the paper does not attempt a comprehensive evaluation of the *performance* of the schemes it investigates. This is certainly a useful and important task, one that is necessary to compare alternatives and ultimately assess the cost and benefits of the different advance reservation services proposed in this paper. However, because the relation between advance reservation and path selection is still relatively unexplored, providing a better understanding of its properties and associated complexity is a useful first step in itself. Such an understanding can then provide the foundation needed for more quantitative performance evaluations.

The rest of this paper is structured as follows. Section II reviews our assumptions regarding the network infrastructure and identifies the range of problems we address. Section III considers a traditional advance reservation model, i.e., users request bandwidth guarantees for some time in the future, and investigates the path selection problem under several scenarios. Section IV is dedicated to a somewhat different model, where advance reservation requests are in terms of the amount of data to be transmitted. Section V considers two further extensions: first, it incorporates delay guarantees; then, within that extension, it discusses a network environment where pricing and preemption priorities are used to differentiate between reservations. Due to space limits, most proofs and technical details are omitted from the paper, and can be found in [9].

II. Infrastructure and Problem Scope

The network is modeled as a graph $G(V, E)$, with a set of nodes V and a set of links E, where $N = |V|$ and $M = |E|$. The time domain, i.e., the domain over which advance reservations can extend, is composed of *time-slots* $\{0, 1, 2, \ldots\}$ of equal size. The duration of a "slot" corresponds to the minimum duration for and advance reservation, and resources are allocated for an integer number of slots, i.e., the duration of the reservation. Consequently, tracking of available resources, e.g., link bandwidth, is also done at the granularity of a slot. In particular, associated with each link $l \in E$, there is a vector $\mathbf{b_l} = \{b_l[0], b_l[1], \ldots,\}$, such that $b_l[t]$ specifies the resources available on link l at time-slot t. Because we focus on connections with bandwidth requirements, $b_l[t]$ denotes the amount of bandwidth available on link l during slot t.

Upon receipt of a new request, the path selection algorithm is invoked to find a feasible path for the request, based on the request requirements and the available network resources specified in the vectors $\{\mathbf{b_l}\}_{l \in E}$. In the case of unicast[1] connections, the connection is established over a *path*[2] between the corresponding source and destination nodes, and resources are reserved on each link of the path for *all* time slots during which the reservation is active. Because the complexity of the path selection process depends in part on the number of time slots that need to be considered, we impose an upper bound T on the maximum completion time of a request.

In the basic advance reservation model, requests are for a given amount of bandwidth between times t_1 and $t_2 \geq t_1$. Path selection then attempts to find a path with sufficient bandwidth in that time interval. Other advance reservation models will translate into different requirements for path selection. For example, the criteria of interest may be to reserve the maximum amount of resources, or to allow for the fastest completion of a task. In such cases, the goal of path selection is not so much to find a path with a sufficient bandwidth between two given time instants. Instead, the "best" path is the one that yields the maximum reserved bandwidth or the earliest end time. As we shall see in Section IV, such variations can result in vast differences in terms of complexity.

Another extension considered in the paper, is the coexistence of advance and standard reservations. In general, one can expect users which pay a higher price, to be given higher priority in the network. For example, advance reservations could be charged more[3] and thus

[1] See [9] for an extension to multicast connections.

[2] [9] also considers the case where up to K paths can be used over the duration of a connection, in an attempt to improve the odds of finding the required resources.

[3] This corresponds to a model where there is a premium associated with the certainty of resource availability in the future. Alternatively, one could envision that advance reservations are made at a discount, as

have the ability to "preempt" existing standard reservations when they become active. Preemption provides a simple mechanism to allow coexistence of different types of reservations, but as discussed in Section V, it also affects the complexity of path selection.

III. Basic Reservation Model

We begin by considering the following basic model. The network is presented with requests, each specifying source and destination nodes, a bandwidth requirement, and potentially a starting time and duration. In other words, a request i specifies a source node $s^i \in V$, a destination node $d^i \in V$, and a bandwidth requirement B^i, which should be reserved for the connection on each link over its path. A request i is also characterized by its starting time t_s^i and duration τ^i, $t_s^i + \tau^i \leq T$; depending on the specific variant of the problem, t_s^i and τ^i may be either fixed or variables. Within the above framework, we consider the following basic problems.

A. Connection Feasibility

Consider first the case of a connection with given bandwidth requirement, starting time, and duration. We want to identify a path that can accommodate the required bandwidth during the specified interval of time. Formally, the problem is as follows.

Problem III-A: *Given a connection request i, with source node s^i, destination node d^i, bandwidth requirement B^i, starting time t_s^i and duration τ^i, find a path $\mathbf{p}$ that supports the bandwidth requirement during the interval $[t_s^i, t_s^i + \tau^i]$.*

The following algorithm provides a solution:

Algorithm III-A:

1. For each link l: if the link does not have B^i units of bandwidth at each time-slot in the range $[t_s^i, t_s^i + \tau^i]$, then delete the link.
2. Considering the residual network, if s^i and d^i are still connected then any connecting path is a solution; otherwise, the connection is not feasible.

Complexity (Algorithm III-A): $O(M \cdot \tau^i)$.

As mentioned earlier, and as seen from the above result, slot granularity, plays a major role in the complexity of the path selection process because of its impact on reservation duration. Increasing slot granularity lowers computation complexity, but at the expense of a potential decrease in performance. This is because resources are allocated for entire slots even when needed only for a fraction thereof. Understanding the trade-off between computational complexity and network performance associated with slot duration is clearly of interest, but outside the scope of this paper. Nonetheless, the computational complexity results derived in this paper provide a starting point for such an investigation.

B. Maximum Duration

Consider now the case in which the duration of a connection is not fixed, and we are interested in prolonging it as much as possible. This could apply in instances where the goal is to maximize the amount of work performed once the task associated with the reservation has started. Formally, the problem is stated as follows.

Problem III-B: *Given a connection request i, with source node s^i, destination node d^i, bandwidth requirement B^i and starting time t_s^i, find a path $\mathbf{p}$ that maximizes the duration of connection i in the range $[0, T - t_s^i]$.*

The solution can be found by first computing, for each link l, the maximum time t_l^i for which l has B^i units of bandwidth available in $[t_s^i, t_l^i]$. The required path is then a widest path with respect to the metric $\{t_l^i\}$.[4] Specifically:

Algorithm III-B:

1. For each $l \in E$:
 (a) $t_l^i \leftarrow t_s^i$;
 (b) while $b_l[t_l^i] \geq B^i$ and $t_l^i < T$ do $t_l^i \leftarrow t_l^i + 1$;
2. Find a path between s^i and d^i that is widest with respect to the metric $\{t_l^i\}$: if there is no such path then the connection is not feasible, otherwise the identified path is the solution.

Complexity (Algorithm III-B): Each t_l^i computation takes $O(T - t_s^i)$ time. The widest path can be computed by conducting a binary search on the value of the solution, each time checking for connectivity through a breadth-first-search (BFS) algorithm [4]. Hence, the total complexity is $O(M \cdot (T - t_s^i) + M \cdot \log(T - t_s^i)) = O(M \cdot (T - t_s^i))$.

As before, slot granularity is a major factor in the computational complexity of the algorithm.

C. Soonest Completion

Consider now the case in which the connection's bandwidth and duration are specified, but not the starting time, and the goal is to identify a path that minimizes the time at which the connection ends, hence also the time at which it starts. This could be useful when the task at hand is a gating factor for the start of a subsequent task. Formally, the problem is specified as follows.

Problem III-C: *Given a connection request i, with source node s^i, destination node d^i, bandwidth requirement B^i and duration time τ^i, find a path $\mathbf{p}$ that enables the earliest starting time t_s^i in the range $0 \leq t_s^i \leq T - \tau^i$.*

in the airline reservation model.

[4] A *widest path* with respect to a metric $\{w_l\}_{l \in E}$ is a path $\mathbf{p}^*$ that maximizes the value of $\min_{l \in \mathbf{p}} w_l$ over all relevant paths $\mathbf{p}$, that is, $\mathbf{p}^* = argmax_{\mathbf{p}} \min_{l \in \mathbf{p}} w_l$.

Algorithm III-C provides a solution. It iterates over all possible starting times, and uses an intermediate variable, t_l^{max}, to record for each link the maximum time known so far, until which link l can support the required bandwidth. Keeping this intermediate variable across iterations avoids having to recompute it every time.

Algorithm III-C:

1. For all $l \in E$: $t_l^{max} \leftarrow 0$.
2. For $t_s^i = 0$ to $T - \tau^i$:
 (a) For each link l:
 • If $t_l^{max} \geq t_s^i + \tau^i$ then do not delete link l during this iteration (i.e., iteration t_s^i).
 • Else ($t_l^{max} < t_s^i + \tau^i$):
 i. If there is not enough bandwidth (B^i) at slot t_s^i at link l, then delete link l during this iteration, and also set $t_l^{max} \leftarrow 0$.
 ii. Else ($t_l^{max} < t_s^i + \tau^i$ and there is enough bandwidth at t_s^i): if $t_l^{max} > 0$ then delete link l during this iteration.
 iii. Else (there is enough bandwidth at t_s^i and $t_l^{max} = 0$): find the maximum interval that can hold the connection, starting at t_s^i, record the last slot in the interval in t_l^{max}; if $t_l^{max} \geq t_s^i + \tau^i$ then do not delete link l during this iteration, otherwise delete it.
 (b) Considering the residual network, if s^i and d^i are still connected *Stop*: any connecting path is a solution.
3. (No path has been identified:) the connection is not feasible.

Complexity (Algorithm III-C): $O(M \cdot (T - \tau^i))$, as each slot at each link is scanned at most twice - once in the "regular" iteration (step 2.(a)), and once in a "look-ahead" step (2.(a)(iii)).

D. Coping with Uncertainties

In [9], we demonstrate how the basic model and solutions described above can be easily adapted to cover several important extensions. In particular, we consider the case where there is uncertainty associated with some of the problem parameters. In the following, we briefly outline some of our findings.

Specifically, we consider cases where either the duration of the requested reservation is not precisely known, or the availability of resources is subject to uncertainty, or both. Such uncertainty is likely to be common, and can be captured by assuming that the parameters are random variables. For example, this is the model of [26, Section B.2], where "call" durations are only approximately known, i.e., in the form of their distribution. Similarly, in [10], uncertainty in available link bandwidth is modeled through the use of random variables, and [1] demonstrates that this is a reasonable assumption. In [9] we assess how this additional dimension affects path selection.

Consider first the basic feasibility problem described in Subsection III-A, but with the duration of connection i being a random variable with probability distribution function (p.d.f.) $q^i(t)$. We seek a path that maximizes the probability of "success", i.e., with enough bandwidth for the duration of the connection; thus, the longer the path can support the required amount of bandwidth, the better it is. Therefore, the problem is identical to the "Maximum Duration" problem considered in Subsection III-B, for which an $O(M \cdot (T - t_s^i))$ solution has been presented.

Assume next, that there is uncertainty with respect to the available bandwidth on a link, with this uncertainty expressed through p.d.f. $\{q_l^t(b)\}_{l \in E}$, which give the probabilities that at a slot t there are at least b units of available bandwidth on a link l. All the variations considered in the previous sections can be extended to accommodate such uncertainty, and solutions can be obtained through simple modifications to those of their deterministic counter-parts. Again, the details can be found in [9].

Nonetheless, some variations can introduce substantial complexity. One such example is when extending the maximum duration problem presented in Subsection III-B, by seeking a path with maximum *expected* duration time. In this specific case, the complexity arises from the difficulty of relating the expected duration of a path to that of its links, and as of now we do not have an efficient solution for this case. Similarly, other extensions can greatly complicate the path selection process, and the next section is devoted to an important such instance.

IV. MAXIMUM TOTAL BANDWIDTH

In this section, we turn to a different service model for advance reservations, that may be of interest in a number of settings. Specifically, suppose that the goal of a request is to transmit a given amount of data, preferably in the minimum amount of time. However, the connection is not constrained to a specific bandwidth value. Instead, it is capable of transmitting at any bandwidth value that is available through the network, and may even be able to adapt to changing bandwidth. In the latter case, the amount of reserved bandwidth can change from slot to slot. In this setting, connection i starting at time t_s^i completes its task at the first time slot $\tau_i \geq t_s^i$, where the sum over all slots $j \in [t_s^i, \tau_i]$ of the product of reserved bandwidth B_j in slot j and the slot duration, exceeds the amount of data that connection i needs to transmit.

Such a service may be desirable for some applications. For instance, a database backup that takes place overnight between a branch office and a server farm. The application will specify a start time, e.g., business closing, and the amount of data it needs to backup, e.g., the size of the database, and will want to make an advance reservation to ensure that the backup completes before the opening of business the next day. The desired bandwidth need not be specified beyond a minimum value as the server will typically be able of adjust its transmission rate based on the available bandwidth. However, because the database and/or the server controlling it may be needed for other maintenance tasks, it is often desirable that the backup complete as quickly as possible once initiated.

The case of a single bandwidth value, i.e., no adaptation from slot to slot, is of limited interest as it is easily reduced to the previous fixed bandwidth value problems. In particular, it can be solved by performing a binary search on the M possible feasible bandwidth values, so that the previous fixed value algorithms can be used. Computational complexity increases only by a factor $\log M$. As a result, we focus on the more interesting case where the reserved bandwidth can change from slot to slot. Unfortunately, as we will see, this seemingly innocuous extension takes us from a simple solution to a hard one.

A. Intractability

As shall be shown, problems belonging to this class of *advance cumulative reservation* (ACR) problems are, in general, intractable. In this paper, because of space constraints, we limit ourselves to showing it for the "basic" case outlined above. A similar result is established in [9], for a more general case where nodes are also allowed to buffer data enroute. Allowing buffering of data allows nodes to shift bandwidth from slot to slot and, therefore, compensate for differences in available bandwidth across links in a given time slot. However, as shown in [9], even this relaxation of the constraints does not help in reducing the complexity of the problem. In general, the different variations of the ACR problem which we have investigated are intractable, as the following result will show.

Consider an elementary instance of the ACR problem, in which the available bandwidth on a link at each time-slot is either "1" or "0". We are presented with a connection request at time $t = 0$, and the question is whether there is a path for the connection, such that during the interval $[0, T]$, the number of time-slots for which all links have a value of "1" is at least β, for some (given) $\beta > 0$. We term this problem the *0-1 Total Bandwidth (0-1 TB) problem.* It is easy to verify that this problem is a special case of the ACR problem. Hence its intractability implies the intractability of the ACR problem.

Theorem IV.1: Problem *0-1 TB* is NP-hard.

Proof: The proof proceeds by transforming a known NP-complete problem into an instance of the ACR problem. Specifically, consider the well known NP-complete SATISFIABILITY (SAT) [8] problem. An instance of the SAT problem consists of a set U of boolean variables and a collection C of clauses over U, i.e., a set of literals over U, for which we are trying to determine if there exists a satisfying truth assignment, i.e., an assignment of the values T (true) or F (false) to the variables in U that simultaneously satisfies all the clauses in C. A clause is said to be satisfied by an assignment iff at least one of its members is true under that assignment.

We now return to the issue of transforming the SAT problem into an instance of the 0-1 TB problem. Consider a SAT problem with clauses C_i, $1 \leq i \leq N$, each composed of literals $\{u_{i_j}\}$, $1 \leq j \leq n_i$, $u_{i_j} \in \{u_k, \bar{u}_k\}$, $1 \leq k \leq K$. We then proceed to construct a network that will allow us to carry out the required transformation. The basic idea is to select a set of nodes and links as well as bandwidth availability at each time slot, in such a way that being able to find a path with a cumulative bandwidth of β units between two nodes requires determining if the answer to the SAT problem is affirmative.

The network topology we use to establish this equivalence is illustrated in Figure 1 for the specific example of $N = K = 3$. It has $K + N + 1$ nodes labeled $0, 1, \ldots, K+N$, where 0 represents the source and $K+N$ the destination. For $1 \leq j \leq K$, there are two links between nodes $j-1$ and j, one (labeled u_j) representing the assignment of T to u_j, while the other (labeled $\bar{u}_j$) represents the assignment of F. In other words, the first K links correspond to the different literals of the problem, and a link will be used to identify the truth assignment selected for its literal in the different clauses of the SAT problem. For $1 \leq i \leq N$, there are n_i links between nodes $K + i - 1$ and $K + i$, each representing one of the literals in $\{u_{i_j}\}$. In other words, the links between node $K + i - 1$ and $K + i$ will be used to capture clause C_i.

Assume now that the time domain over which reservations can be made on each link has $\sum_{k=1}^{N} n_k$ slots. As we will see, slot number $(\sum_{k=1}^{i-1} n_k + j)$, $1 \leq j \leq n_i$, will be used to determine if "clause C_i is satisfied by u_{i_j}". The bandwidth availability vectors of the various links are then set as follows:

- For link u_j: for $1 \leq i \leq N$, $1 \leq m \leq n_i$, if $u_{i_m} = \bar{u}_j$, then slot number $(\sum_{k=1}^{i-1} n_k + m)$ is set to 0, otherwise it is set to 1. This amounts to setting each slot to 0 or 1 depending on whether the literal in the corresponding clause has a value of $\bar{u}_j$ or not.
- For link $\bar{u}_j$: for $1 \leq i \leq N$, $1 \leq m \leq n_i$, if $u_{i_m} = u_j$, then slot number $(\sum_{k=1}^{i-1} n_k + m)$ is set to 0, otherwise it is set to 1. Similarly, this amounts to setting each slot to 0 or 1 depending on whether the literal in the corresponding clause has a value of u_j or not.
- For a link between nodes $K + i - 1$ and $K + i$, representing the literal u_{i_j}: slots $1, 2, \ldots, (\sum_{k=1}^{i-1} n_k)$ are set to 1; slots $(\sum_{k=1}^{i-1} n_k + 1), \ldots, (\sum_{k=1}^{i-1} n_k + j - 1)$ are set to 0; slot $(\sum_{k=1}^{i-1} n_k + j)$ is set to 1; slots $(\sum_{k=1}^{i-1} n_k + j + 1), \ldots, (\sum_{k=1}^{i} n_k)$ are set to 0; slots $(\sum_{k=1}^{i} n_k + 1), \ldots, (\sum_{k=1}^{N} n_k)$ are set to 1.

The above choices can be explained as follows: the "literal" links correspond to an assignment of T or F to a literal; a link representing u_{i_j} "filters" all literals of C_i but u_{i_j} (such a link can be referred to as a "filtering edge for C_i"). This is best understood in the context of the example of Figure 1, which we describe shortly.

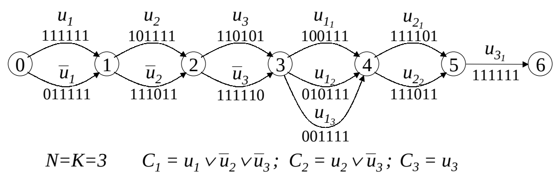

Fig. 1. Network Topology for Instance of SAT Problem.

Based on the above construction, it can be verified that there is a path between the source and destination with total bandwidth of $\beta = N$ units iff the answer to the SAT problem is affirmative. □

As mentioned above, an example is helpful to clarify the proof. Specifically, Figure 1 shows the network topology and link bandwidth for an instance of the SAT problem with $N = K = 3$, and the following three clauses: $C_1 = u_1 \vee \bar{u}_2 \vee \bar{u_3}$, $C_2 = u_2 \vee \bar{u}_3$, and $C_3 = u_3$. As shown in [9], it is easy to verify that the SAT problem has the unique solution $u_1 = u_2 = u_3 = T$, and that the corresponding 0-1 TB problem, for $\beta = 3$, has the unique solution, i.e., path, $\{u_1, u_2, u_3, u_{1_1}, u_{2_1}, u_{3_1}\}$.

B. Approximations

In view of the intractability of Problem 0-1 TB, a natural next step is to consider approximations. We begin with an approximation for the basic 0-1 TB problem, and then turn to consider the more general case, where link bandwidths may take values other than 0 and 1. The basic idea behind the approximation is to eliminate from consideration paths whose *expected* number of slots with non-zero bandwidth is significantly below the amount β we are seeking. In the 0-1 case, this amounts to focusing on the expected number of 1's on a path.

B.1 0-1 Bandwidth Values

This is again limited to an environment where bandwidth can only take the values of either 0 or 1. In order to allow us to determine the expected number of 1's on a path, we also make the following assumption.

Assumption U1: the probability that a slot on a link has a value of "1" is q, $0 < q < 1$, independently of the values on other slots and links.

We note that assumption *U1* ignores the possible correlation implied by multi-slot reservations, and also assumes an homogeneous load in the network. Both factors can clearly introduce discrepancies from this idealized model. However, a global estimate of temporal link congestion such as q can provide a first order estimate of the likelihood of finding the needed number of "1" slots on a path. Consider now some path $\mathbf{p}$, with m hops. We are interested in finding the expected number of "1" slots in the path, i.e., slots that have a 1 value at all m links. Denote that value by K_m.

Proposition IV.1: Under Assumption U1, $K_m = T \cdot q^m$.

A similar result can be obtained under a slightly different assumption on how "1" slots are distributed.

Assumption U2: each link has K slots with the value of "1"; their distribution is uniform and independent of the distributions on other links.

While assumption $U2$ may appear less artificial than assumption $U1$, it is essentially identical.

Proposition IV.2: Under Assumption U2, $K_m = K \cdot \left(\frac{K}{T}\right)^{m-1}$.

Note that using $q = K/T$ gives identical values for K_m in propositions IV.1 and IV.2.

Because propositions IV.1 and IV.2 are equivalent, we focus on Assumption $U1$ and the corresponding Proposition IV.1 in the rest of this section.

Our approximation is based on the previous observation that a path, whose expected number of "1" slots is significantly below the required amount β, will usually not be a solution and hence can be neglected. As explained in the following, together with Proposition IV.1, this leads to the conclusion that we can limit our search to paths whose number of hops is below a fixed, small, value. Specifically, suppose we disregard paths whose expected number of "1" slots is below $\alpha \cdot \beta$, where α is some fixed number, smaller than 1 (e.g., $\alpha = 0.1$). Then, given the value of q corresponding to Assumption $U1$, the maximal number of hops that we should consider is at most $m_{\max} = \frac{\log(\frac{T}{\alpha \cdot \beta})}{\log \frac{1}{q}}$.

Hence, a reasonable approximation is to search all paths of at most $m_{\max}$ hops, and check their quality, i.e. number of "1" slots. Denoting by D the maximal node degree, such a search incurs a complexity of

$$O(D^{m_{\max}}) = O(D^{\frac{\log(\frac{T}{\alpha \cdot \beta})}{\log \frac{1}{q}}}). \qquad (1)$$

The above expression indicates that complexity can be high for extreme values of β and q. However, in such cases one can defer to rather trivial alternatives, whose use can be triggered by a simple check on β and/or q.

1. When the value of β is small, say $\beta = \delta$, $\delta << T$, a simple solution is provided by searching all possible combinations for choosing δ "1"-slots out of a total of T slots. For each combination so obtained, suitable paths can then be obtained by (i) removing the links that have a "0" value in any of the corresponding δ slots and (ii) checking connectivity on the residual network; the complexity is $O(T^{\delta+1} \cdot M)$.

2. When q is close to 1, most paths are likely to satisfy our hop count limitation, so that searching through all of them would be very inefficient as is reflected in expres-

sion (1) (for q close to 1). A possible alternative, that works well when the value of β is neither small nor large, is to simply choose paths at random[5] and check their quality – under the above conditions, the expected number of tests till a "good enough" path is found would be small. This will, however, not work well when β is close to T as even with q close to 1, a random selection may produce too many misses to yield a feasible path.

3. However, when the value of β is close to T, say $\beta = T - \delta$, $\delta << T$, because the number of possible "misses" (i.e., zero slots) one can afford is small, an exhaustive search can again be contemplated. Specifically, one can search all possible combinations of δ misses, and for each combination: (i) remove the links that have a "0" value in any of the remaining $T - \delta$ slots, and (ii) check connectivity on the residual network. The complexity for this case is again $O(T^{\delta+1} \cdot M)$.

Hence, the search over paths of at most $m_{\max}$ hops can be limited to the cases where q is not too close to 1 and β is neither too close to nor significantly smaller than T. Coupling these observations with the fact that, typically, in communication networks D grows only slowly with the network size N (e.g., $D = O(\log N)$) and, moreover, is often bounded (i.e., $D = O(1)$), we conclude that the complexity of the proposed approximation as given in expression (1) should be reasonably small.

To illustrate this, suppose that $\alpha = 0.1$, $D = \log_2 N$, and we seek the minimal value of β which would render the complexity to be lower than $O(N^2)$. Considering $q = 0.5$ and topologies of at least moderate size, say $N \geq 100$, it is sufficient for β to be just as large as $\frac{T}{291}$, whereas for $q = 0.9$ it is sufficient for β to be just as large as $\frac{T}{16}$. For either larger topologies (i.e., $N > 100$) or smaller values of q, the results would be even better (i.e., smaller allowable values of β). As a second example, suppose we consider $\alpha = 0.1, N \geq 100, T \geq 100$, and compare the complexity expression (1) with $O(M \cdot T)$, which, from what we have seen, is typically a lower bound on the complexity of advance reservation schemes: for $q = 0.5$, the minimal value of β, for which (1) is lower than $M \cdot T$, is $\frac{T}{582}$, whereas for $q = 0.9$ the value is below $\frac{T}{18}$.

We can, therefore, conclude that for topologies of at least moderate size, values of q which are not too close to 1, and values of β that are not very small with respect to T, the approximation performs better, in terms of computational complexity, than $O(N^2)$. The penalty in terms of performance amounts to neglecting paths whose (a priori) expected quality is below one tenth of the required quality β. In addition, as mentioned before, the approximation relies on several idealized assumptions, that may not always hold in practice and which would, therefore, also affect overall complexity and/or efficiency. Furthermore, and most important, while the above approximation provides a possible handle on what we identified as a hard problem, it does so only for the simple case of 0-1 bandwidth values. A meaningful approximation would have to tackle the more general problem of arbitrary bandwidth values. We briefly review such an extension in the next sub-section, and the reader is referred to [9] for details.

[5] Possibly using some heuristic to get closer to the destination.

B.2 General Bandwidth Values

In this subsection, we outline how to extend the approximation obtained for the 0-1 case to handle general bandwidth values. In other words, we consider the case where link bandwidths can take values in the range $[1, \ldots, B]$, for some $B > 1$, and propose an approximation based on the assumption that we can obtain an "optimal" solution for the 0-1 case, as described above.

The basic idea behind the approximation is to quantize the link bandwidths into a finite set of values, and for each value convert the problem into a 0-1 problem by assuming that the available bandwidth is 0 when it is less than the selected value and 1 otherwise. The approximation then selects the best solution from those generated by the 0-1 approximations for each bandwidth values. Due to space limits, the details are omitted, and the reader is referred to [9]; there, an algorithm is presented, which achieves an approximation factor of $\frac{1}{e \cdot lnB}$ over a solution to the basic 0-1 problem.

To summarize, in this section we have shown that although an advance reservation model that focuses on cumulative (total) bandwidth might be an attractive service extension, its impact on the complexity of the path selection process is substantial. In particular, we established the intractability of a basic instance, namely the 0-1 TB problem. In light of this result, we investigated possible approximations and provided a reasonably efficient solution. The solution relies on a number of assumptions, and as such needs to be further evaluated in more practical settings; nevertheless, it provides an initial step towards addressing what turned out to be hard problem.

In the next section, we consider several other extensions and enhancements to the service model of advance reservations, and again investigate their impact on the path selection.

V. Delay Constraints, Priorities, and Preemption

In this section, we consider two other important extensions to the basic advance reservation service model. First, we extend the basic model to incorporate delay guarantees in addition to bandwidth guarantees. Then, for this general service model, we discuss a network environment where pricing and priorities are the mechanisms used to integrate support for both advance and immediate reservations. In particular, reservations are assigned priorities that determine the order in which they can be *preempted* in case of insufficient resources. As discussed in [26], "over-booking" of resources can occur for numerous reasons, e.g., uncertainty in reservation durations, and preemption provides a simple yet effective mechanism for resolving such contentions. However, preemption and the associated use of priorities impact the path selection process, which now needs to be aware of not only link reservation levels, but also their corresponding priority.

A. Delay Constraints

Consider first the extension where the advance reservation service model allows connections to specify not only minimum bandwidth guarantees, but also an end-to-end delay bound for their packets. In order to concretize the discussion, we limit our attention to the case where delay guarantees are in terms of a hard upper bound. Specifically, we consider the *rate-based* service model of [24] for providing end-to-end delay bounds. Under this model, and in the framework of advance reservations, links and connections are characterized in the following way.

Each link $l \in E$ is characterized as follows:

- A vector[6] of the bandwidth (or *rate*) $\mathbf{b_l} = \{b_l[t]\}_{t=0}^{T-1}$, which the link can offer to new connections at each timeslot t.
- A *constant delay* value δ_l, related to the link speed, propagation delay, and maximum transfer unit size.

An advance reservation request i is characterized by the following parameters:

- A *source node* s^i and a *destination node* d^i.
- A *bias* value σ^i, i.e., the connection's burst size.
- A *maximum packet size* c^i.
- A *maximum end-to-end delay constraint* D^i.
- A *bandwidth requirement* B^i.
- Start time and duration t_s^i and τ^i, correspondingly.

When the request i is routed over a path $\mathbf{p}$ with a reserved rate r, $r \leq \min_{l \in \mathbf{p}, t \in [t_s^i, t_s^i+\tau^i)} b_l[t]$, the following upper bound $D^i(\mathbf{p}, r)$ on the end-to-end delay applies:

$$D^i(\mathbf{p}, r) = \frac{\sigma^i + n(\mathbf{p}) \cdot c^i}{r} + \sum_{l \in \mathbf{p}} \delta_l \qquad (2)$$

where $n(\mathbf{p})$ is the number of hops on $\mathbf{p}$. We then consider the following basic feasibility problem in the context of path selection.

Problem V-A: Given are (1) an advance reservation request i, and (2) a network characterized as described above, choose a path $\mathbf{p}$ (with $n(\mathbf{p})$ hops) between s^i and d^i, and a rate r^i, such that:

1. $r^i \geq B^i$;
2. for all $t_s^i \leq t < t_s^i + \tau^i$, and for all $l \in \mathbf{p}$: $b_l[t] \geq r^i$;
3. $\frac{\sigma^i + n(\mathbf{p}) \cdot c^i}{r^i} + \sum_{l \in \mathbf{p}} \delta_l \leq D^i$.

[6]When we later introduce priorities, there will be a separate vector for each priority class.

A simple solution to this problem is provided by the following algorithm.

Algorithm V-A:

1. For all links l: $r_l \leftarrow \min_{t_s^i \leq t < t_s^i + \tau^i} b_l[t]$.
2. For all ($O(M)$) different values of $r \in \{r_l\}$, such that $r \geq B^i$:
 (a) Delete all links l for which $r_l < r$;
 (b) for each (remaining) link l, compute: $w_l \leftarrow \frac{c^i}{r} + \delta_l$;
 (c) compute (on the residual network) a path $\mathbf{p}(r)$ that is shortest with respect to $\{w_l\}$;
 (d) If $\frac{\sigma^i}{r} + \sum_{l \in \mathbf{p}(r)} w_l \leq D^i$ then *stop*: $\mathbf{p} = \mathbf{p}(r)$ and $r^i = r$ are a solution.
3. (No solution has been identified:) The connection is not feasible.

Complexity (Algorithm V-A): step (1) incurs $O(M \cdot T)$, while step (2) can be done in $O(M \cdot (N \log N + M))$, thus resulting in an overall complexity of $O(M \cdot (T + N \log N + M))$.

B. Priorities and Preemption

In this subsection, we introduce pricing and priorities as means to differentiate between different types of reservations, so that higher priority reservations can preempt lower priority ones. We articulate the discussion within the context of general bandwidth and delay guarantees.

We consider an environment with J *service classes*, such that, for $1 \leq j \leq J$, a connection belonging to class j can preempt connections of classes $j+1, \ldots, J$. The bandwidth available in class j on link l can be expressed as a time dependent vector, i.e., $b_l^j[t]$, $0 \leq t \leq T$. The ability to preempt lower priority reservations implies that:

$$b_l^1[t] \geq b_l^2[t] \geq \ldots \geq b_l^J[t], \ \ 0 \leq t \leq T-1; \qquad (3)$$

this is because a higher priority request "sees" as available the bandwidth reserved for connections in lower priority classes, which it can preempt.

Similarly to [15], there is a (time-dependent) cost function for establishing a connection over a path of n hops and with reserved rates $r^1, r^2, \ldots, r^J$ in each priority class. The cost function is of the form:

$$C(t, n, r^1[t], \ldots, r^J[t]) \ \ 0 \leq t \leq T-1,$$

where $r^j[t]$ denotes the reserved rate for the j-th class traffic at time slot t. In addition, we make the following assumptions:

1. $C(t, n, r^1[t], r^2[t], \ldots, r^J[t])$ is nondecreasing in n and in $r^j[t]$, $1 \leq j \leq J$. In other words, reserving more bandwidth in any class increases the overall cost, and so does increasing the hop count (number of links) of the paths where the reservation is made. The latter reflects the fact that, generally, the cost to the network of a given

reservation, increases with the number of links on which the required bandwidth needs to be set aside.

2. For all i, j and r, $i < j$ and $r > 0$:

$$C(t, n, r^1, \ldots, r^i, \ldots, r^j + r, \ldots, r^J) < C(t, n, r^1, \ldots, r^i + r, \ldots, r^j, \ldots, r^J). \quad (4)$$

This simply states that the cost of bandwidth increases with the priority class where the bandwidth is to be reserved.

A connection request i is characterized as before and by a set of end-to-end delay bounds D^j, one for each priority class j, $1 \leq j \leq J$. We further require that the delay bounds requested in each priority class by a connection, be invariant across time slots. This fits a setting where the different bounds correspond to various levels of performance that are feasible for the application. The goal is then to minimize, over the life-time of the connection, the cost of the associated reservations. As a result, and while different rates could be allocated in different time slots, the goal of minimizing the path cost implies that, within a given class, the reserved rate is constant from slot to slot. Furthermore, because it is not practical to assume that user packets can be replicated and sent over multiple paths, we require that the delay bounds specified for each priority class all be achieved on the *same* path $\mathbf{p}$. In accordance to equation (2), the delay bound D^j of class j is then associated with a corresponding bandwidth reservation r^j in each class on all links of $\mathbf{p}$. Finally, because of our goal to minimize cost, the reserved rates[7] are the smallest possible in each class, so that:

$$r^j[t] = r^j = \frac{\sigma + n(\mathbf{p}) \cdot c}{D^j - \sum_{l \in \mathbf{p}} \delta_l}, \quad (5)$$

for $j = 1, 2, \ldots, J$, $t_s^i \leq t < t_s^i + \tau^i$. Because of the higher cost of bandwidth in high priority classes and the relation expressed in equation (5) between delay and rate, the delay bounds for each priority class and the corresponding rates can be assumed to be ordered as follows:

$$D^1 \geq D^2 \geq \ldots \geq D^J, \quad r^1 \leq r^2 \leq \ldots \leq r^J. \quad (6)$$

In other words, the above ordering states that a user can only "afford" looser delay bounds (smaller rates) in higher priority classes. For example, a user may ask for a relatively large (but tolerable) delay bound D^1 in the highest priority class, in order to minimize (avoid) the risk of being totally shutoff (preempted) by other requests, while containing the cost of such a guarantee. However, in order to enjoy better performance (lower delay) whenever possible, the user may also request a tighter delay bound $D^j < D^1$ in a lower priority class. This will require a higher reserved bandwidth r^j than the class 1 reservation r^1, but the associated cost increment will be small because of the lower cost of bandwidth in priority class j. This model enables a user to tailor its reservation based on performance requirements, tolerance to performance degradations, and ability/willingness to pay. Equations (3), (5), and (6) imply that it suffices to check the feasibility of rate reservations $\{r^1, r^2, \ldots, r^J\}$ on a path $\mathbf{p}$ just with respect to the last class J, i.e., whether $r^J \leq \min_{l \in \mathbf{p}, t_s^i \leq t < t_s^i + \tau^i} b_l^J[t]$. Hence, the following algorithm finds a feasible path of minimum cost.

Algorithm V-B:

1. For all links l, compute the residual rates $R_l^J \leftarrow \min_{0 \leq t \leq T} b_l^J[t]$.
2. Order the different values of the residual link rates R_l^J by decreasing order: $R(1) \geq R(2) \geq \ldots \geq R(K)$, where $K \leq M$.
3. For $k = 1$ to K:
 (a) Delete links l with $b_l^J \leq R(k)$.
 (b) Run a Bellman-Ford shortest-path algorithm in order to find, for all hop values n, $1 \leq n \leq H$, a path $\mathbf{p}(k, n)$ that is shortest with respect to $\{\delta_l\}$ among paths of at most n hops.
 (c) For all $1 \leq n \leq H$:
 i. If $\frac{\sigma^i + n(\mathbf{p}(k,n)) \cdot c^i}{D^J - \sum_{l \in \mathbf{p}(k,n)} \delta_l} > R(k)$ or $\sum_{l \in \mathbf{p}(k,n)} \delta_l \geq D^J$ then $\mathbf{p}(k, n) \leftarrow null$;
 ii. else : for all $1 \leq j \leq J$: $r^j(k, n) \leftarrow \frac{\sigma + n(\mathbf{p}(k,n)) \cdot c}{D^j - \sum_{l \in \mathbf{p}(k,n)} \delta_l}$.
4. Among all paths $\mathbf{p}(k, n)$, $1 \leq k \leq K$, $1 \leq n \leq H$, choose a path $\mathbf{p}$ that minimizes $C(n, r^1(k, n), \ldots, r^J(k, n))$.

Complexity (Algorithm V-B): $O(M \cdot (T + H \cdot (M + J)))$.

The above algorithm illustrates one instance, namely a fixed delay bound (and bandwidth guarantee) for each priority class across all time slots, for which the additional complexity of advance reservations is marginal[8]. However, as with the standard model (i.e., bandwidth guarantees only and no preemption), there are cases where introducing advance reservations appears to drastically affect the complexity of the path selection. One likely example is when different delay bounds are allowed in each time slot, possibly to reflect changes in the cost of bandwidth from slot to slot. This would correspond to a model where delay bounds are not directly associated with application performance levels, but instead are chosen on the basis of what the user can afford. It appears that this problem is also intractable, but formally establishing this result is the topic of ongoing work.

[7] For simplicity, we assume that the reserved rates in each class are larger than the required bandwidth guarantees. Removing this constraint is easily achieved through a simple modification of step 3(b)ii of Algorithm V-B.

[8] Step 1 of the algorithm removed the dependency on time, so that from Step 2 and on it is essentially the algorithm used in the case of "immediate" reservations.

VI. CONCLUSION

The goal of this paper was to provide an initial look at how advance reservations affect the complexity of the path selection process, as it is a topic that had been left largely untouched. There are many aspects of path selection that are impacted by the introduction of advance reservations, and this paper has certainly ignored many, focusing primarily on the issue of computational complexity. This is by no means an indication of the lesser significance of other issues such as, for example, network performance; rather, it reflects a choice of starting point given the limited amount of work done so far on the topic. In that context, we believe that some of the initial results provided in this paper, establish some of the required foundations for ultimately understanding the complex interactions between advance reservations and path selection.

In particular, we have seen cases where advance reservations had only a minor effect on path selection, while in other cases it turned simple problems into intractable ones. For example, the initially promising extension that considered advance reservations aimed at cumulative (variable) bandwidth values, is likely to be impractical because of its intractability. On the other hand, several simple extensions, e.g., maximum duration, soonest completion, etc., were found to be easily dealt with through relatively simple algorithms. In general, we feel it is important to carefully assess the cost of providing new advance reservation services, so that this can be compared to their potential benefits. Determining the added computational complexity introduced by advance reservations is one of the important elements of such an overall cost assessment. The results of this paper are, therefore, meant as contributions towards this goal. In addition, the paper also identified several new extensions whose complexity remains an open question, e.g., time varying delay bounds. Answering those questions is the topic of ongoing work, so as to better assess the potential usability of such extensions.

REFERENCES

[1] G. Apostolopoulos, R. Guérin, S. Kamat, and S. Tripathi. Improving QoS routing performance under inaccurate link state information. In *Proceedings of ITC'16*, Edinburgh, Scotland, June 1999.

[2] A. Banerjea, D. Ferrari, B.A. Mah, M. Moran, D.C. Verma, and H. Zhang. The Tenet real-time protocol suite: Design, implementation, and experiences. *IEEE/ACM Trans. Netw.*, 4(1):1–10, February 1996.

[3] R. Braden (Ed.), L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource reSerVation Protocol (RSVP) version 1, functional specification. Request For Comments (Proposed Standard) RFC 2205, Internet Engineering Task Force, September 1997.

[4] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, Cambridge, MA, 1990.

[5] M. Degermark, T. Kóhler, S. Pink, and O. Schelén. Advance reservations for predictive service. In *Proceedings of NOSSDAV'95*, Durham, NC, April 1995.

[6] L. Delgrossi and L. Berger (Eds.). Internet STream protocol version 2 (ST2), protocol specification - version ST2+. Request For Comments (Experimental) RFC 1819, Internet Engineering Task Force, August 1995.

[7] D. Ferrari, A. Gupta, and G. Ventre. Distributed advance reservation of rel-time connections. In *Proceedings of NOSSDAV'95*, Durham, NC, April 1995.

[8] M.R. Garey and D.S. Johnson. *Computers and Intractability*. Freeman, San Francisco, 1979.

[9] R. Guérin and A. Orda. Networks with advance reservations: The routing perspective. Technical report(s), University of Pennsylvania & Department of Electrical Engineering, Technion, June 1999. Available from: http://pender.ee.upenn.edu/~guerin/publications/go99.ps & ftp://ftp.technion.ac.il/pub/supported/ee/Network/go99.ps.

[10] R. Guérin and A. Orda. QoS-based routing in networks with inaccurate information: Theory and algorithms. *IEEE/ACM Transactions on Networking*, 7(3):350–364, June 1999.

[11] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured Forwarding PHB group. Request For Comments (Proposed Standard) RFC 2597, Internet Engineering Task Force, June 1999.

[12] V. Jacobson, K. Nichols, and K. Poduri. An Expedited Forwarding PHB. Request For Comments (Proposed Standard) RFC 2598, Internet Engineering Task Force, June 1999.

[13] Y. Liang, K.-Q. Liao, J. W. Roberts, and A. Simonian. Queueing models for reserved setup telecommunication services with advance capacity reservation. In *Proceedings 12th Intl. Teletraffic Congress*, Torino, Italy, June 1988. Paper 4.4B.1.

[14] K.-Q. Liao and J. W. Roberts. Videoconference traffic and network design. *IEEE. Trans. Commun.*, COM-35(3):275–282, March 1987.

[15] A. Orda. Routing with End to End QoS Guarantees in Broadband Networks. *IEEE/ACM Transactions on Networking*, 7(3):365–374, June 1999.

[16] W. Reinhardt. Advance reservation of network resources for multimedia applications. In *Proceedings of 2nd Intl. Workshop on Advanced Teleservices and High-Speed Communication Architectures (IWACA'94)*, Heidelberg, Germany, September 1994.

[17] W. Reinhardt. Advance resource reservation and its impact on reservation protocols. In *Proceedings Broadband Islands'95*, Dublin, Ireland, September 1995.

[18] J. W. Roberts. Teletraffic models for the Telecom 1 integrated services network. In *Proceedings 10th Intl. Teletraffic Congress*, Montreal, P.Q., Canada, June 1983. Paper 1.1.2.

[19] J. W. Roberts and K.-Q. Liao. Traffic models for the telecommunication services with advanced reservation. In *Proceedings 11th Intl. Teletraffic Congress*, Kyoto, Japan, June 1985. Paper 1.3-2.

[20] O. Schelén and S. Pink. An agent based architecture for advance reservations. In *Proceedings of 22nd Annual Conf. Local Comput. Netw. (LCN'97)*, Minneapolis, MN, November 1997.

[21] O. Schelén and S. Pink. Sharing resources through advance reservations agents. In *Proceedings of IWQoS'97*, New York, NY, May 1997.

[22] A. Schill, F. Breiter, and S. Kühn. Design and evaluation of an advance resource reservation protocol on top of RSVP. In *Proceedings of IFIP Broadband'98*, Stuttgart, Germany, April 1998.

[23] A. Schill, S. Kühn, and F. Breiter. Resource reservation in advance in heterogeneous networks with partial ATM infrastructures. In *Proceedings of INFOCOM'97*, pages 612–619, Kobe, Japan, April 1997.

[24] S. Shenker, C. Partridge, and R. Guérin. Specification of Guaranteed Quality of Service. Request For Comments (Proposed Standard) RFC 2212, Internet Engineering Task Force, September 1997.

[25] J. Virtamo. A model of reservation systems. *IEEE. Trans. Commun.*, 40(1):109–118, January 1992.

[26] D. Wischik and A. Greenberg. Admission control for booking ahead shared resources. In *Proceedings of INFOCOM'98*, San Francisco, CA, April 1998.

[27] J. Wroclawski. Specification of the controlled-load network element service. Request For Comments (Proposed Standard) RFC 2211, Internet Engineering Task Force, September 1997.