A Policy-Based Service Specification for Resource Reservation in Advance

Martin Karsten[1], Nicole Beriér[2], Lars Wolf [1], and Ralf Steinmetz[1,2] *

1
KOM - Industrial Process and System Communications
Darmstadt University of Technology
Merckstr. 25
64283 Darmstadt
Germany
Email: {Martin.Karsten,Lars.Wolf,Ralf.Steinmetz}@KOM.tu-darmstadt.de

2
GMD-IPSI.Mobile
German National Research Center for Information Technology
Dolivostr. 15
64293 Darmstadt
Germany
Email: {Nicole.Berier,Ralf.Steinmetz}@darmstadt.gmd.de

Abstract

Resource reservation in advance can be a useful extension for a reservation-based communication service, if certain users are willing to pay additional charges to overcome the blocking probability of a communication network. Existing proposals for advance reservations restrict service flexibility by technically enforcing a certain subset of service invocations and conceptual separation of immediate and advance requests. In this paper, we present the specification of a single general network service providing both immediate and advance reservations. We introduce a policy and pricing layer to calculate compensations for certain service characteristics. Correct operation of this service is shown and further extensions are discussed. Employing such a communication service and the respective policy layer, users are free to choose the characteristics of their service requests as long as the system is able to deliver all given guarantees. However, the system is controlled by a policy layer, specifically, users are subject to charges, which vary according to their service parameters.

* This work is sponsored in part by: Volkswagen-Stiftung, Hannover, and Deutsche Telekom AG, Darmstadt.

1 Introduction

For communication services implemented by means of resource reservation, capacity planning should be based on economic calculations incorporating estimations of price elasticity and expected demand. However, there is always a possibility of demand exceeding the available resources. In such a case, some reservation requests have to be rejected in order to guarantee correct handling of others. This possibility is referred to as *blocking probability*. Traditional telephone networks are usually dimensioned to keep the blocking probability very low by overprovisioning resources compared to estimated demand patterns based on well-established experience. For future integrated services networks, precise demand patterns might be harder to estimate, due to the greater flexibility of usage requests. Therefore, to achieve a very low blocking probability, additional overprovisioning would be necessary. One might argue that sharing resources between best-effort and reservation-based services lowers the need of overprovisioning resources. This is correct, but only to a certain extend, because a certain amount of best-effort traffic that has to be transmitted by a network might be isolated and shielded as well, to provide an acceptable level of service. It is largely speculative to predict the application-mix and consequently the level of multiplexing in future integrated services networks.

The concept of resource reservation in advance allows to specify reservation requests ahead of time [1]. Such requests only make sense, if their blocking probability is smaller compared to that of an immediate reservation. In practice, advance reservation requests might only be used, if their blocking probability is zero, except in case of hardware failures. One particular application are mobile devices moving from one access point to another and establishing a reservation in advance to continuously keep the communication service alive [2,3]. Especially in the area of wireless communication, transmission capacity is a crucially limiting factor, hence, it might not be possible to guarantee a very low blocking probability. In general, candidates for advance reservations are timed communication requests of such high importance that the normal blocking probability is not acceptable (e.g. telemedicine). Given these aspects, resource reservation in advance can be considered as an additional management technique to coordinate shared usage of limited resources. In this paper, we provide a service definition and policy layer to exploit these benefits. Thereby, we present a new approach to advance reservation by separating the management into capacity and policy decisions. We intentionally use a generic description and do not tie our approach to a certain network technology.

The rest of the paper is organized as follows. In Section 2, we review and assess related work. We present a unique and general service definition in Section 3 and the complementary policy layer in Section 4. In Section 5, we demonstrate the general applicability of our approach by further extending the flexibility of service invocations. Finally, we summarize and conclude the paper and briefly present our ideas for future research work in Section 6.

2 Related Work

A number of approaches to resource reservation in advance have been published so far. Many of these concentrate on the issue of enabling advance reservation in the first place and signalling appropriate requests between network nodes. The fundamental problem of resource reservation in advance is depicted in Figure 1. Given a certain amount of future requests and no limitation on service duration, is it possible to schedule incoming reservations?

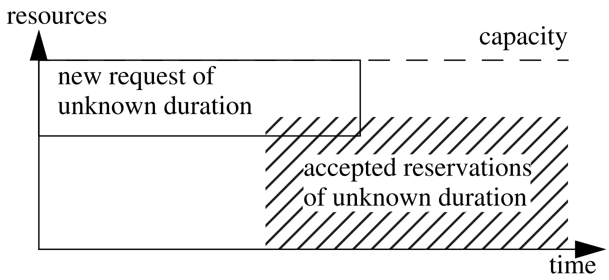

Figure 1: Scheduling of Advance Reservations

The work presented in [4] indicates that occasionally preempting exisiting reservations in favour of adavance reservations can increase overall resource utilization. In [5], an agent-based reservation system is presented, in which immediate and advance reservations are handled differently. Advance reservations always have to specify a finite duration and are never preemptable. Immediate reservations never specify a duration and are always preemptable. The system considers certain time horizons, called *lookahead time* and *bookahead time*, to decide about acceptance of immediate and advance reservations. To us, this service model introduces unnecessary limitations, which are of questionable virtue. For example, the authors note that selection of the time horizons is crucial for useful operation and certain requests are inherently precluded. On the other hand, users are expected to pay for service requests, so the question remains why certain requests (which are complicated for the system to handle) should be completely prohibited, instead of just setting appropriately high charges. Different handling of advance and immediate requests is introduced as an architectural benefit, however, we believe it only adds complexity to the system. The system is further described, evaluated and implementation details for admission control are given in [6].

A different approach [7] suggests that advance reservations also specify a service duration, but immediate reservations are not preemptable. Admission control for reservation requests in advance is done by only considering other advance reservations. Advance and immediate reservations are isolated by dynamically partitioning the network resources. Because the partition for advance reservations has to be large enough to admit all requested future reservations, this might lead to a situation, in which a significant amount of resources cannot be assigned to immediate requests, yet being unused.

In [8], an architecture for realizing advance reservations in an IP/RSVP-based network is suggested and discussed.

For the RSVP policy framework, the relevant proposed standardization document [9] defines priority levels for service preemption. However, no background on advance reservation and the task of assigning priority levels is given. Further details on the exact signalling procedures for enabling advance reservations on top of RSVP can be found in [10]. Other approaches we are aware of, but do not discuss here for reasons of brevity, include [11,12]

Basically all previously suggested approaches conceive the fundamental admission control problem associated with resource reservation in advance. However, the attempts to deal with and completely solve this problem by technical means usually fall short, because of the limited scope of such approaches. One particular problem is given by the strict conceptual separation of immediate and advance reservations and the requirement to specify the duration of an advance reservation. As a consequence, this irrevocably limits the service time.

Realizing this, we present an integrated and generic service definition for immediate and advance reservations and delegate part of the admission control problem to a policy layer. This service definition does not fundamentally deviate from other suggestions, but our model is significantly less complex, yet more general, in that functionality at the network layer is restricted to essential aspects, thus being very simple.

3 Network Service

We aim to specify a uniform reservation-based service description that covers both immediate and advance reservations. The service description should impose as few restrictions as possible on potential service requests. On the other hand, each router must be able to determine whether a pending request can be accepted without violating guarantees given to other reservation requests.

When accepting an advance reservation, there is a *hold-back time*, the timeframe between service request and service invocation. The fundamental problem when accepting advance reservations can be formulated as follows: During the hold-back time, how can the allocated resources be used for other requests? If other advance reservation requests arrive, which specify a service duration, it can be determined whether these are schedulable. A more difficult situation is given with immediate reservations, because usually these do not specify a fixed service duration. Three basic solutions exist for this problem. The first is that each reservation request, including immediate reservations, also specifies a duration and is only accepted if resource availability can be guaranteed for the whole duration. The second possibility is to preempt service requests when their resources are needed for an advance reservation. As a third alternative, resources could be partitioned for immediate and advance reservations, such that no preemption is needed and only advance reservations have to specify a duration. It can be concluded from previous research efforts (see Section 2) that at least one of these solutions has to be adopted by the network. However, we feel that there is no need to technically restrict the system to either one.

Partitioning of resources should be avoided if possible, because it prohibits resource sharing. Even in case of dynamic partitioning [7], future advance reservations block resources for other immediate reservations. Declaring the duration of service invocations might not be possible and acceptable for all users and usage scenarios. On the other hand, the possibility of preemption might also not be acceptable under all circumstances. Therefore, we specify a network service that does not rely on partitioning and integrates both preemption and duration declaration in a general way. Nevertheless, the potential for precisely predicting service guarantees is retained. We achieve this goal by distinguishing between duration of non-preemptable service and actual reservation lifetime.

3.1 Service Definition

A service request for a resource reservation R is described at request time by the 4-tuple (r,s,e,v) as follows:

- r: time of reservation request
- s: begin of service
- e: end of non-preemptable service
- v: amount of resource capacity

That is, at time r, a user requests an advance reservation of capacity v, starting at time s, which is guaranteed not to be preempted until time e. This description does not include the actual service duration, which can be arbitrary. The difference of s and r expresses the hold-back time for an advance reservation. The key supplement to this service description is the following specification: At each time t, each service request is in a state p(t), called *preemption priority*, with

$$p(t) = \begin{cases} 1 & s \le t < e \\ 0 & \text{else} \end{cases} \qquad (1)$$

If p(t) = 1, then the reservation request is guaranteed not to be preempted. A reservation is assigned a preemption priority of 1 for the time that is specified in the reservation request. At the end of this duration the reservation is not automatically torn down, instead it is just considered preemptable for the sake of scheduling other non-preemptable requests. Employing this additional state description, the flexibility for requesting and managing advance reservations is extended, because even if a duration has to be specified for non-preemptable reservations, this does not necessarily result in a fixed a-priori reservation time. This service definition is graphically depicted in Figure 2. We give some examples to demonstrate the flexibility of this service definition, each requesting an arbitrary amount of resources v:

- immediate and preemptable reservation at time t_0: $R(t_0,t_0,t_0,v)$
- advance reservation at time t_0 for time t_1 requesting a minimum service time l: $R(t_0,t_1,t_1+l,v)$
- immediate reservation at time t_0 requesting a minimum service time k: $R(t_0,t_0,t_0+k,v)$

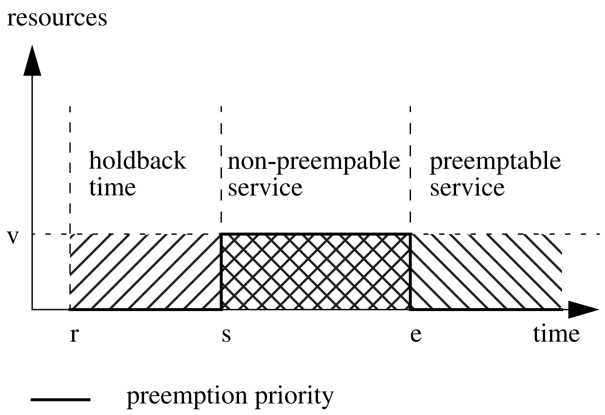

Figure 2: Service Definition

Using this service definition, each possible instantiation of immediate and advance reservations combined with the choice of preemption priority can be requested. The service definition is independent of the actual duration of reservation, it only determines the amount of time when a reservation request is not to be interrupted. It turns out that considering non-preemptable time is sufficient to define a general reservation service.

3.2 Admission Control

In order to provide service guarantees for blocking probability and preemption, an admission control algorithm is needed. For this algorithm, only non-preemptable service requests have to be considered at each time, because all other reservation requests can be preempted. In this sense, we first define the total load at time t:

$$\text{load}(t) = \sum_{i=1}^{n} v_i \cdot p_i(t)$$

for p_i,v_i from all service requests R_i, $i = 1,...,n$ (2)

Defining C as total capacity and t_0 as current time, a set of reservations R_i, $(i = 1,...,n)$, is schedulable, iff

$$\text{load}(t) \le C \qquad \text{for all } t,\ t \ge t_0 \qquad (3)$$

We now intuitively show how to use this definition as an admission control condition and then formalize its usage. Consider the situation shown in Figure 3. The dotted line denotes the total available capacity of resources and the long-dashed line depicts the current time t_0. The dashed line represents existing and requested non-preemptable reservations. At time t_0, two new advance reservation requests arrive, one of which is schedulable while the other one is not. For an immediate request, non-preemption can be guaranteed for a certain amount of time. Preemptable reservations are not shown in this figure, because they do not influence the calculation of overall schedulability of non-preemptable requests. We conclude this section by formally specifying the admission control condition:

At time t_0, a new service request $R_x = (t_0,s_x,e_x,v_x)$ can be accepted by the system, iff

$$\text{load}(t) + v_x \le C \qquad \text{for all } t,\ s_x \le t < e_x \qquad (4)$$

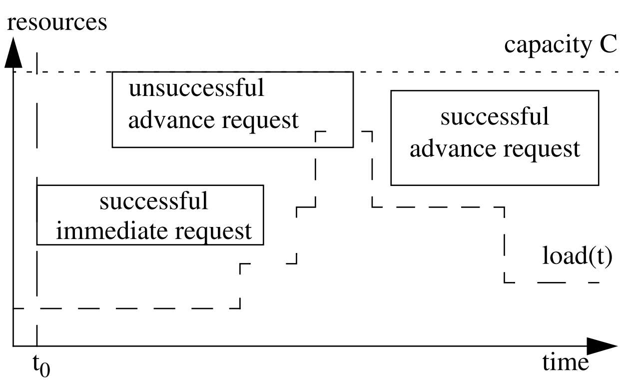

Figure 3: Existing and New Advance Reservations

For admission control, it is sufficient to consider those times at which load(t) changes its value. If we denote these times with τ_j, a simple algorithmic description can be given as follows:

```
decision = Accept
for each τj, sx ≤ τj < ex
   if (load(τj) + vx) > C
   then decision = notAccept
endfor
```

Note that this admission control condition does not principally differ from those of existing proposal, it just considers a subset of existing reservations only. Therefore, proposals to implement such an admission control algorithm, as for example the work presented in [6], can be applied here, as well.

3.3 Service Invocation

There are several ways of invoking this service with a signalling protocol. One possibility would be to use a handshake mechanism:

user → system: REQUEST(s,v)
system → user: RESPONSE(e_{max})
user → system: CONFIRM(e) or REFRAIN

The user requests a certain amount of resources at time s and the system responds by specifying the maximum duration this reservation can be guaranteed to be non-preemptable. Then, the user either confirms requesting the service by choosing an end time or refrains from service invocation.

However, a handshake mechanism like this inhibits the problem that additional overhead is needed to keep the decision an atomic one. State information and timers would be needed to detect hanging invocations. Therefore, we propose the following protocol elements to invoke the reservation service:

user → system: REQUEST(s,e,v)
system → user: ACCEPT or REJECT(e_{max})

When using this service, a user specifies start time s, end time e and an amount of resources v. The system responds by either accepting the request or rejecting it, depending on its current state. In case the service is rejected, the system announces the currently possible maximum duration for non-preemptable service on an *informational* basis, i.e., without guarantees. This information can be used by the end-system to adapt its requirements and issue a new request. Additional information might be added in case of service rejection, for example, an alternative start time. This service invocation model is idempotent and atomic and therefore, significantly reduces the complexity of protocol implementation. It also nicely integrates with RSVP's one-pass mechanism for reserving resources [13].

4 Policy Layer

As discussed in the previous sections, there are fundamental conflicts associated with the admission control problem for advance reservations. To us it is clear that a general solution to this problem cannot be found, especially not purely in the network layer. Therefore, we approach this issue by delegating the decision about acceptance of advance and preemption of existing reservations to a policy layer. In general, resource reservation introduces discrimination between usage requests and therefore, a policy layer is needed to control, coordinate and compensate for resource consumption in the first place.

4.1 General Aspects

Several constraints can be identified when a policy scheme for advance resource reservation is being developed. We already briefly discussed the issue of protocol implementation in Section 3.3. Using a policy layer requires a network node to actually make two decisions atomically (admission control & policy control) which increases complexity. This overhead is bound, because the service specification employs a very simple invocation model.

It seems to be an open question whether advance reservations should be subject to additional charges or receive a discount. Advance reservations increase complexity in the network, however, a network provider can extract planning information for the future, which can be economically useful. To decide whether an advance reservation should be given a rebate or charged an additional fee largely depends on the ability to adapt a network's capacity to demand, i.e., the planning horizon. If a reservation request is received, which reserves resources after a certain point in time and if the sum of all reservation requests are significant enough to adapt capacity, it is potentially suitable to grant a discount for this request. However, such a discount is currently beyond the scope of our model, because many other externalities would have to be considered as well, for example: trust in the user, time of payment, general market developments, etc.

Advance reservations and specification of preemptable and non-preemptable service time create additional means of discrimination between usage requests, therefore, compensation can be demanded from users requesting such features. We consider an immediate and preemptable reservation to be "normal" and suggest to charge an increased fee for further service characteristics. Although preemption is an integral part of our service model, in real operation we consider it an exceptional condition that does not occur regularly, because of careful capacity planning. Under this assumption, an alternative suggestion for pricing would be the airline model of overbooking aircraft seats. This could be applied by charging the same price for preemptable and non-preemptable reservations and in case of preemption, a compensation would be paid by the network provider.

We now formulate requirements to a pricing model for the basic scheme, considering the service definition from Section 3.1. The effort of holding an advance reservation increases with the amount of time it is booked ahead, because other requests are potentially blocked. Consequently, the charge for a reservation request should positively correlate to its start time s compared to request time r, i.e. (s-r). Similarly, a positive correlation should apply for the duration of non-preemptable service (e-s) and its price. Such a pricing model additionally serves as barrier against highly problematic requests, without completely prohibiting them in the network layer. For example, a request for an infinite duration of non-preemptable service is not excluded in the service definition, but given a positive correlation, it results in an infinitely high price. As another example, a reservation request in advance specifying no non-preemptable service duration provides no benefit to the user, but nevertheless requires management effort by the system. Hence, a higher price than for an immediate reservation should apply to discourage users from such requests.

The pricing model we propose in the following section is mainly intended to provide information for internal calculation of a network provider. In particular, prices only denote those parts of the total price which are resource-dependent. A separate fixed flow setup charge might apply, which in combination with resource-based components leads to the usual characteristic that the function of price per resource unit is sub-additive for an increasing amount of resources. Such a fixed fee would cover the fixed costs per flow setup, for example for state maintenance in the system. Actual sale prices may further deviate from calculatory prices because of marketing and other general considerations.

4.2 Pricing Model

In order to derive prices for service requests, an a-posteriori service description is needed, which includes an additional parameter d, expressing the actual duration of resource reservation. The service description for a request R is then given by the 5-tuple (r,s,e,v,d). The price consists of three components reflecting actual resource usage by reservation and scheduling effort for advance respectively non-preemptable reservations. Following the justification in [14], we assume all price components to be linear in the amount of resources. Considering the requirements listed in the previous section, the price function looks as follows:

$$p(r, s, e, v, d) = a_1 \cdot v \cdot d + a_2 \cdot v \cdot (s - r) + a_3 \cdot v \cdot (e - s) \tag{5}$$

The first component expresses plain resource consumption during the actual reservation time. The second component accounts for the hold-back time of an advance reservation, and the third component includes non-preemptable service time into price calculation. Note that all addends in this formula depend on the amount of resources v that is being reserved. This is due to the fact that for each price component the corresponding amount of effort is correlated to the amount of resources. The coefficients a_1, a_2 and a_3 are subject to economic calculation of a network provider, which is beyond the scope of this paper. We briefly explain how the requirements from Section 4.1 are fulfilled by this price formula:

- The price positively correlates to the hold-back time and for an infinite time, the price becomes infinite through the second component.
- The price positively correlates to the non-preemptable service time and for an infinite time, the price becomes infinite through the third component.
- A “useless” request for an advance reservation without any non-preemptable service is still subject to additional charges through the second component.

Our claim is that the generic service model in conjunction with this pricing approach provides a high flexibility yet reasonable control of immediate and advance reservation requests. Our approach essentially precludes infinite reservations by making them infinitely expensive. If desirable, it could be combined with a partitioning approach as in [7] in a sense that a (dynamically sized) partition is priced differently.

5 Service Extension

In this section, we demonstrate the general applicability of our approach to advance reservations by extending the service definition and also embedding this extended service model into a modified policy layer. The service extension is done by allowing to modify the non-preemptable duration of an existing reservation request. In that sense, a modification can be classified (see Figure 4) by the fact whether the new non-preemptable service duration is completely covered by the previous selection (case 1) or not (case 2). If yes, no special action has to be performed at the network layer, whereas otherwise admission control has to be executed on the modified request. However, both cases require activity in the policy layer.

5.1 Modified Service Request without Admission Control

As a specific example, we examine the concept that users are allowed to reduce the amount of non-preemptable service time by lowering the end time parameter e. This can formally be reflected by an additional parameter e’:

e’: modified end of a non-preemptable service request, with e’ < e

In order to cover this extended service by a policy layer, the a-posteriori service description has to be extended, as well. Besides including e’ into the service description, the time of this modification request is important, because the earlier

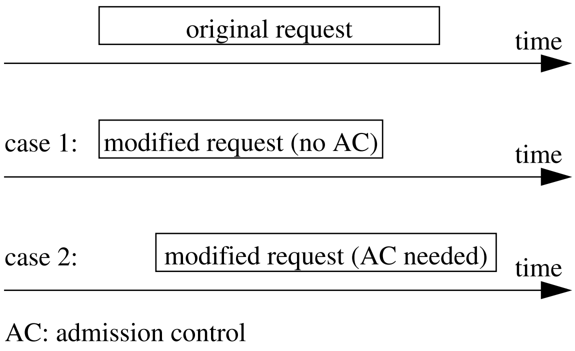

Figure 4: Modification of reservation requests

the non-preemptable service time is reduced the more benefit (from better scheduling potential) the system has.

m: modification time of a service request

Given the above considerations, the discount for such a modification should depend on both e’ and m. It should not affect the price components for resource consumption and hold-back time from (5), but only the surcharge for non-preemption, i.e. the third price component from (5). A discount formula has to adhere to some other requirements as well:

- if m = r, the discount should cover the whole surcharge
- the discount should never exceed the surcharge
- if m = e, the discount should be zero
- the discount should never fall below zero

Using (5) as a basis, we can express the discount as follows:

$$\mathrm{discount}(r, e, e', m) \; = \; a_3 \cdot v \cdot \left(b_1 \cdot \left(1 - \frac{m - r}{e - r}\right) + b_2 \cdot \left(\frac{e - e'}{e - r}\right)\right)$$

$$\text{with } b_1 + b_2 = 1 \qquad (6)$$

The last factor (*discount factor*) of this formula determines the discount in relation to the original surcharge and consists of two components. The expression multiplied by b_1 denotes the influence of *when* the request is modified, while the expression multiplied by b_2 describes by *how much* the non-preemptable time is reduced. The coefficients b_1 and b_2 allow weighting both aspects. The discount factor varies between 0 and 1. This discount formula satisfies all requirements listed above. A similar formula can be derived for deferring the start time without modifying the end time of non-preemptable service.

5.2 Modified Service Request with Admission Control

If admission control is needed for a modified service request, it has to be treated differently by the system. For admission control, the existing request has to be taken into account, such that it is not counted twice. Since admission control might fail, it seems most appropriate to consider this as a new service request. With respect to policy control, this is suitable as well, because the existing request can be deleted and charged, applying the discount calculation of the previous section. In case of acceptance, a new price for the modified request can then be calculated from scratch.

6 Summary and Future Work

In this paper, we have discussed the fundamental admission control problem associated with resource reservation in advance. Our conclusion from previous related work is that none of the approaches is flexible enough to cover all potential needs of all users. By separating the issue into a technical and a policy part, we are able to specify a generic service description and a corresponding policy layer, in particular, appropriate pricing formulas. The combination of both improves flexibility compared to other approaches, yet retaining reliable and precise admission control.

Many issues remain open for further research work. The proposed reservation model has to be verified, simulated and tested to back up the hypothesis of its advantages. We will continue to work on this issue, especially in the area of realizing our approach in combination with existing reservation protocols, for example using an implementation of RSVP [15]. As well, applications of advance reservations in the area of mobile networking promise interesting results [3]. On the theory side, it is still an open question, which reservations should be preempted, if necessary and if there is a choice. In case of very scarce resources this could be investigated by means of economic auctions. Last not least, we only provided the general structure of a parameterized pricing formula. A calculation framework is needed to actually derive price coefficients or a completely different price function.

References

[1] L. C. Wolf, L. Delgrossi, R. Steinmetz, S. Schaller, and H. Wittig. Issues of Reserving Resources in Advance. In *Network and Operating System Support for Digital Audio and Video, Fifth International Workshop, NOSSDAV'95, Durham, New Hampshire, USA*. Springer LNCS 1018, April 1995.

[2] A. T. Campbell. The Mobiware Toolkit. In *Fourth International Workshop on High Performance Protocol Architectures (HIPPARCH'98), London*, June 1998. Invited Paper.

[3] A. Pajares, N. Berier, L. Wolf, and R. Steinmetz. An Approach to Support Mobile QoS in an Integrated Services Packet Network. In *First Workshop on IP Quality of Service for Wireless and Mobile Networks (IQWiM'99). Aachen, Germany.*, April 1999.

[4] A. G. Greenberg, R. Srikant, and W. Whitt. Resource Sharing for Book-Ahead and Instantaneous-Request Calls. *IEEE/ACM Transactions on Networking*, 1(7), February 1999.

[5] O. Schelén and S. Pink. An Agent-based Architecture for Advance Reservations. In *Proceedings of IEEE 22nd Annual Conference on Computer Networks (LCN'97), Minneapolis, USA*, November 1997.

[6] O. Schelén, A. Nilsson, J. Norrgard, and S. Pink. Performance of QoS Agents for Provisioning Network Resources. In *Proceedings of the 7th IEEE/IFIP International Workshop on Quality of Service (IWQoS'99), London, UK*, pages 17–26. IEEE/IFIP, June 1999.

[7] D. Ferrari, A. Gupta, and G. Ventre. Distributed Advance Reservation of Real-Time Connections. In *Network and Operating System Support for Digital Audio and Video, Fifth International Workshop, NOSSDAV'95, Durham, New Hampshire, USA*. Springer LNCS 1018, April 1995.

[8] S. Berson, R. Lindell, and R. Braden. An Architecture for Advance Reservations in the Internet. Technical report, USC Information Sciences Institute, July 1998. Available at http://www.isi.edu/%7eberson/advance.ps.

[9] S. Herzog. Preemption Priority Policy Element. Internet Draft: draft-ietf-rap-signaled-priority-01.txt, January 1999. Work in Progress.

[10] A. Schill, F. Breitner, and S. Kühn. Design and Evaluation of an Advance Reservation Protocol on top of RSVP. In *IFIP 4th International Conference on Broadband Communications, Stuttgart, Germany*, pages 23–40. Chapman & Hall, April 1998.

[11] W. Reinhardt. Advance Reservation of Network Resources for Multimedia Applications. In *2nd International Workshop on Advanced Teleservices and High Speed Communications Architectures (IWACA'94), Heidelberg, Germany*, October 1994.

[12] M. Degermark, T. Köhler, S. Pink, and O. Schelén. Advance Reservations for Predictive Services. In *Network and Operating System Support for Digital Audio and Video, Fifth International Workshop, NOSSDAV'95, Durham, New Hampshire, USA*. Springer LNCS 1018, April 1995.

[13] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. RFC 2205 - Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. Proposed Standard, September 1997.

[14] M. Karsten, J. Schmitt, L. Wolf, and R. Steinmetz. Provider-Oriented Linear Price Calculation for Integrated Services. In *Proceedings of the 7th IEEE/IFIP International Workshop on Quality of Service (IWQoS'99), London, UK*, pages 174–183. IEEE/IFIP, June 1999.

[15] M. Karsten. KOM-RSVP Protocol Engine, 1999. Work in Progress. Software available from http://www.kom.e-technik.tu-darmstadt.de/rsvp/.