Proceedings of KiVS'99 (Kommunikation in Verteilten Systemen), Darmstadt March 1999

Cost and Price Calculation for Internet Integrated Services

Martin Karsten[1], Jens Schmitt[1], Lars Wolf[1], and Ralf Steinmetz[1,2] *

1: Technische Universität Darmstadt
Merckstr. 25 • 64283 Darmstadt

2: GMD IPSI
Dolivostr. 15 • 64293 Darmstadt

Email: {Martin.Karsten,Jens.Schmitt,Lars.Wolf,Ralf.Steinmetz}@KOM.tu-darmstadt.de

Abstract Charging schemes are needed to protect an integrated services network from arbitrary resource reservations and to create a funding mechanism to extend network capacity at the most desired locations at the expense of those users that actually use these resources. While not being the only input into pricing and charging, cost calculation is an important part of a charging scheme. In this paper, we develop a technique called *virtual resource mapping* to apply well-known economic principles to an optimal pricing framework and other tasks related to charging. Additionally, we describe how *virtual resource parameters* can be used to express prices when being combined with protocol mechanisms for charging. We focus on rate-based service guarantees in the context of Internet Integrated Services (IntServ) combined with IP multicast and RSVP as signalling protocol. It turns out that under given aggregated price-demand patterns, resource costs can precisely be extracted for each service request. Thereby, virtual resource parameters can be considered as link between economic theory and technical reality.

Keywords Charging, Cost Calculation, Pricing, Rate-based QoS, Integrated Services.

1 Introduction

The transition of the Internet towards a commercially funded and used integrated services network raises, among others, the question about how network usage can be charged appropriately. Clearly, current charging schemes (mainly flat-fee access-based or time/volume-based) will not be sufficient in the presence of multiple service classes, resource reservations and discrimination between different usage requests [MMV97]. From an economic point of view, communication services are characterized by:

- availability of a non-storable resource (network capacity)
- high fixed costs & low variable costs

In economic theory, these characteristics, which are similar to traditional telephony, electricity, aircraft seats, etc., are dealt with by using a management technique called *Yield Management* [Lei98]. When Yield Management is used, prices are not calculated using full-cost or variable-cost based calculation. Instead, prices are highly differentiated depending on the expected demand. In the context of communication networks, granting a reservation request is profitable as long as the charge for this request is higher than its marginal cost. However, to reach the optimum profit, opportunity costs must be added to the marginal costs, i.e., a resource reservation prohibits using the resource for another request with a potentially higher revenue. In fact, opportunity costs dominate marginal costs by far, since variable costs are negligibly low. The main task is to optimize capacity and prices according to a given price elasticity (i.e. demand per price), such that the overall revenue is maximized. In [WPS97], a framework for optimal pricing and capacity planning of a generic guaranteed services network is given.

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

In this work, we assume the existence of a known aggregated price-demand function and therefore, knowledge about the optimal capacity and expected aggregated demand. We concentrate on the issue of cost calculation and allocation to service requests for different service classes. In order to keep cost calculation and pricing practically tractable, it is desirable to characterize resource usage by comparable parameters. Moreover, other charging tasks like cost allocation and protocol issues can be simplified by using comparable resource parameters or even a single generic parameter, as well. Opposite to this requirement, reservation requests for communication services in a multiple service class network are usually described by a multi-dimensional QoS vector, containing, for example, peak and average bandwidth and end-to-end delay, which cannot easily be compared between different service classes.

We focus on Internet Integrated Services (IntServ) [BCS94] and describe how to handle an actual reservation request containing a multi-dimensional flow specification. Our main contribution is a method to compare reservation requests and extract precise resource costs. Thereby, the practical use of an existing optimal pricing framework [WPS97] is simplified. We also briefly describe other fields of employment [FD98, HSE97]. Finally, we show how resource costs can be used for price representation using RSVP charging mechanisms as described in [KSWS98].

The structure of this paper follows the outline above. After discussing related work in Section 2, we discuss the IntServ service classes with respect to resource usage in Section 3. Afterwards, in Section 4, virtual resource mapping and a cost model using virtual resource parameters is described. We then show how to use this cost model for various calculation approaches in Section 5 and present a protocol related use in Section 6. In Section 7, we summarize our results and give an outlook to further research issues.

2 Related Work

The problem of charging for network communication can be split in multiple, partially interdependent aspects. In this section, we briefly consider existing work on these aspects.

Calculation Cost and price calculation provides the economic background for setting charges. Most of the currently available literature about charging considers economic aspects of network communication by seeking price models to optimize the overall welfare of all users [MMV95, SFY95, GSW95, KVA98, CSKW98] or the network provider's profit [WPS97]. While being very valuable, these approaches essentially represent an application of previously existing knowledge from economic theory to idealized or very general networking scenarios.

Protocol Calculation and charging is hardly possible based only on local knowledge, therefore protocol definitions are necessary to exchange charging related information between network entities. In [FSVP98, KSWS98, CSZ98], suggestions for defining protocol elements are made with differing levels of detail. It is important to realize the novel challenge for charging protocol elements, opposite to existing data communication technology: transmission of a protocol message might cause an immediate obligation to pay charges, therefore protocol definitions not only need to be functionally correct, but also must a concise definition of their legal semantics.

Architecture A charging architecture composes all charging components, including calculation, protocol aspects and billing. It is important that these components fit together, for example, charging protocol elements must carry all information necessary to set a price. In [SCEH96], the *Edge Pricing* paradigm was identified to be a crucial feature of any charging architecture. Furthermore, a charging architecture must be developed having in mind that any assumption about cooperation between network entities is not valid anymore when individual payment obligations are the consequence of participating in a charging mechanism.

This paper is focused on applying economic results on calculation of costs and prices to existing network technology. However, we also consider how our method can be used with regard to protocol-related aspects of charging. Similar work has been carried out in [CSKW98], but major differences exist. In [CSKW98], the underlying traffic model is based on the notion of effective bandwidth, which is a statistical value, and only considers a single service class. Furthermore, it is not stated how the results can be used by a charging architecture. In this paper, we highly simplify the problem by exploiting existing definitions of service classes and implicitly using the underlying worst-case oriented network calculus of the IntServ framework. Thereby, the work in [CSKW98] can partly be considered as more general, but also as less applicable by having a different foundation and direction.

3 Resource Usage of IntServ Service Classes

The IETF's IntServ framework [BCS94] defines services classes for reservation-based QoS provisioning in IP networks. Currently, the *Controlled Load* [Wro97] and the *Guaranteed* [SPG97] service classes are in the process of standardization. Because of its complementary relation to Guaranteed service, we additionally consider the proposed *Guaranteed Rate* [GGPR96] service class in this paper. We also feel that extending the set of service classes is useful to show the general applicability of our model.

In the IntServ framework, RSVP [BZB+97] is used as control protocol to carry reservation requests and IntServ-enabled routers install reservations to discriminate among different data flows to guarantee a certain level of service to each of them. The full flexibility of the receiver-oriented and anonymous IP multicast model as well as the inherent robustness of a connectionless network protocol can be exploited by using this approach. In the following, while briefly reconsidering the IntServ service classes, we specify their properties with respect to resource usage.

3.1 Controlled Load Service

The definition of Controlled Load service is somewhat fuzzy, in that a traffic flow, characterized by a token bucket, receives a network service similar to best-effort service under "lightly loaded conditions". An imprecise service definition like this is highly unsuitable for commercial network services in the first place, because, as many authors point out, a charging scheme for transmission services requires a well-defined quality definition and measurable performance objectives [KSWS98, FD98, Asa98, Gal97]. While [Wro97] states implementation and evaluation guidelines for Controlled Load service, there are still a number of implementation options left open. For different implementations, slightly different resource usage patterns can be expected, however, all of them have important aspects in common:

- The required service rate can occasionally exceed the token bucket rate.
- The required buffer can occasionally exceed the burst-capable buffer.

Both resources (especially the excessive parts) can be subject to pooling between multiple flows, as long as the probability of excessive loss or delay is fairly low.

3.2 Guaranteed Service

Guaranteed service is intended for applications that have stringent worst-case delay requirements, for example on-line conferencing or distributed interactive simulations. A traffic flow, characterized by a token bucket, receives its requested service rate at each router. If the service rate is enforced for all routers along a flow's path, a bound on the end-to-end delay can be guaranteed for all packets belonging to this flow as has been shown in [PG93, PG94]. This service can be implemented in several ways. A straightforward implementation uses weighted fair queuing (WFQ) [DKS89] to guarantee the service rate. Other approaches suggest to use a combination of traffic shaping and deadline-based scheduling [GGPR96] to obtain lower buffer requirements and jitter bounds, although this increases the average end-to-end delay.

From an economic point of view, there are some interesting aspects related to Guaranteed service. First, while tighter delay bounds result in a higher service rate, they actually reduce buffer requirements. Second, when a reservation for Guaranteed service is issued, it is distinguished between the token rate of the traffic description and the service rate which eventually determines quality of service. Usually, there is a difference between both, the sum of which (over all G flows) can be used to provide another service class, called *Guaranteed Rate* in [GGPR96]. The accumulated differences between token and service rate of all Guaranteed service flows can on the other hand also be used as the rate pool that is needed to provide the excess service rate for Controlled Load (see Section 3.1). However, in [DVR98], it is shown how careful setting of both values affects the end-to-end delay, which could lead to reservation requests where the token rate equals the service rate. Therefore, appropriate charging must provide an incentive to keep the token rate as low as possible yet reflecting the actual average data rate. We consider this by having separate cost components for token rate and service rate. At this point, we do not consider the optional *slackterm* parameter of a Guaranteed service QoS specification. It has no direct influence on cost and price calculation, because its usage only indirectly affects setting of other service parameters.

3.3 Guaranteed Rate Service

As mentioned in Section 3.2, the delay guarantees of Guaranteed service are actually achieved by overreserving a certain service rate, which however, remains unused most of the time. Therefore, [GGPR96] and others suggested to define the Guaranteed Rate service to make use of these unused resources in a more controlled fashion then by best-effort traffic. The semantics are a long-term guarantee about an average transmission rate, but no guarantees about the end-to-end delay. The underlying assumption of proposing this service is that even if there were not much demand for it in the first place, it might be possible to sell it that cheap that customers are attracted by it.

4 Resource Mapping for IntServ Service Classes

In this section, we first explain why buffer usage can be neglected for resource costs of IntServ service classes. Afterwards, we formulate a model to map the remaining rate parameters onto virtual resource parameters and use those to handle cost and price calculation.

4.1 Eliminating Buffer Consideration

It turns out that the IntServ service classes' resource usage can be described basically by rate and buffer parameters. As mentioned above, a router is not required to use per-flow rate-based scheduling, however, the rate-based semantics of the IntServ classes suggest that an implementation's resource usage will be similar to this scenario. We consider the buffer-to-rate ratio of service requests by dividing the required buffer space by the service rate. Even for a very large and bursty traffic stream, this ratio remains at approximately 1 $\frac{\text{MB}}{\text{Mbit/s}}$ (see appendix). In general, we expect that the quotient of both will hardly ever exceed 10 $\frac{\text{MB}}{\text{Mbit/s}}$. Therefore, we compare this number with real investment costs.

We (over)estimate the current price for memory with roughly US$ 5 per MB. The price of a leased line at OC-3 speed (155 Mbit/s) is assumed to be more than US$ 50000 per month plus a per-mile distance charge, while discounts up to 50% are possible (see [Lei98, FO98] and references herein). To handle a buffer-to-rate relation of 10 $\frac{\text{MB}}{\text{Mbit/s}}$ in an OC-3 interface, the amount of buffer needed is 1550 MB, which is equivalent to US$ 7750. Expecting 3 years of equipment usage and only US$ 25000 as monthly line costs, the total costs of buffer are still less than 1% of the total costs for the leased line. While we are aware that these costs will decrease over time, we in principle assume that the relation between buffer and link costs will remain roughly the same as with the current cost structure.

The conclusion from this observation is obvious: If it is feasible to equip an outgoing interface with sufficient buffer space, such that queuing buffer will never really become a bottleneck and if this buffer equipment comes at 1% of the link costs, then it is perfectly legitimate to neglect resource usage of buffer space for cost calculations.

4.2 Virtual Resource Mapping

In reality, only one resource parameter (service rate, i.e., forwarding capacity) denotes the rate resource of an outgoing link. However, there are up to two rate parameters, R and r, in IntServ service specifications with even different semantics depending on the actual service class. In order to allocate costs to reservation requests, we therefore establish a cost model using three *virtual resource parameters*, on which the IntServ rate parameters are mapped.

- The *token rate* (q_T) describes the forwarding rate that is always available and expected to be constantly used by a flow.
- The *clearing rate* (q_C) denotes a guaranteed forwarding rate on top of the token rate that is reserved per delay-guaranteed flow, but expected to be used only for bursts of data.
- The *residual rate* (q_R) is a forwarding rate on top of the token rate, which is only statistically available to a flow. This resource represents the unused capacity of q_C.

Using these parameters, mapping the *R* and *r* parameter from a flow specification to the virtual resource parameters is done according to Table 1:

service class	q_T	q_C	q_R
Guaranteed	r	R - r	-
Controlled Load	r	-	e
Guaranteed Rate	-	-	r

Table 1: Resource allocation for IntServ service classes

In Table 1, parameter *e* denotes the additional rate that is needed to support the occasional excess needs of Controlled Load service. Calculation of this parameter depends on the token bucket specification of a service request and is mainly dependent on the actual implementation choice for Controlled Load.

Our goal is to find a linear function

$$cost(x_T, x_C, x_R) = \mathrm{a}x_T + \mathrm{b}x_C + \mathrm{c}x_R \quad (1)$$

to assign resource costs to a flow requesting token rate x_T, clearing rate x_C and residual rate x_R. Costs are applicable per fixed time unit, which can be chosen arbitrarily small. In such a model, the time parameter is a constant scaling factor, therefore we do not explicitly consider it for the rest of this section.

4.3 Cost Model

When using Yield Management, a cyclic dependency (shown in Figure 1) exists between the various calculation steps. The following cost model is not intended to be a complete solution for the task of setting prices, but it is an important piece of this cyclic process. We artificially break the cycle by assuming the existence of a known price-demand curve for aggregated resource usage of each resource in each service class.

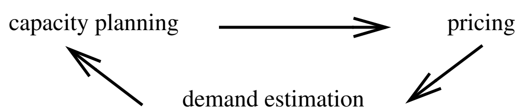

Figure 1: Cyclic dependency among calculation tasks

That given, it is possible to determine the optimum prices and provide capacity according to the demand, therefore we can calculate the expected demand and revenue for each resource parameter. Note that in reality it is usually not possible to estimate the correct price-demand curve, instead an approximation can be generated based on experience of past measurements. In this case, it is highly useful to only measure aggregated parameters. The expected demand can be mapped on the virtual resource parameters as well, hence we know the following revenue parameters:

Let $D_X(S)$ be the aggregated demand and (2)

let $rev_X(S)$ be the aggregated revenue (3)

for service class $S \in \{G, CL, GR\}$ and virtual resource q_X, $X \in \{T, C, R\}$ with

G: Guaranteed service, CL: Controlled Load, GR: Guaranteed Rate and

q_T: token rate, q_C: clearing rate, q_R: residual rate.

Of course, the corresponding accumulated values are known, as well:

$$rev(S) = rev_T(S) + rev_C(S) + rev_R(S) \text{ for service class } S \in \{G,CL,GR\} \text{ and} \quad (4)$$

$$D(S) = D_T(S) + D_C(S) + D_R(S) \text{ for service class } S \in \{G,CL,GR\} \quad (5)$$

We only consider opportunity costs which are equal to the respective price for each unit of virtual resources. To be precise, the coefficients a, b and c denote costs *and* price per resource unit. Therefore, using (1), (2), (4) and knowing the empty fields in Table 1, we can establish the following revenue equations:

$$rev(G) = a \times D_T(G) + b \times D_C(G) \quad (6)$$

$$rev(CL) = a \times D_T(CL) + c \times D_R(CL) \quad (7)$$

$$rev(GR) = c \times D_R(GR) \quad (8)$$

Solving these equations produces the coefficients for the cost function, as well. As the last step, the service-specific cost functions result from mapping the virtual resources back to the original parameters:

$$cost_G(r,R) = cost(r,R\text{-}r,0) = a \times r + b \times (R\text{-}r) \quad (9)$$

$$cost_{CL}(r) = cost(r,0,e) = a \times r + c \times e \quad (10)$$

$$cost_{GR}(r) = cost(0,0,r) = c \times r \quad (11)$$

Depending on the context, it might be desirable to calculate a fraction of total costs for a service request, instead of calculating an absolute cost value. Using opportunity costs, this can be achieved by dividing the absolute cost value by the total revenue.

5 Application to Calculations

5.1 Optimal Pricing

The authors of [WPS97] present a very general and complete model for optimal pricing of multiple guaranteed service classes under consideration of price-demand functions. It is correctly pointed out that analytically solving the whole model is mathematically intractable, therefore a heuristic procedure is described to apply the results. While other research approaches often deal with optimal pricing in a sense of optimal welfare [MMV95, SFY95, GSW95, KVA98CSKW98], this pricing scheme is targeted to maximize profit for the provider. However, as noted in [WPS97], a similar model can be developed to maximize other objectives. Furthermore, any model can benefit from virtual resource parameters. We simplify the general model and apply virtual resource mapping for IntServ service classes in several ways:

- Instead of using very general assumptions about admission control and the properties of service classes, we exploit the knowledge about IntServ service classes to make requests for different classes comparable.
- We do not explicitly consider a spot market for best-effort traffic, because first, we do not believe this to be technically achievable and second, it is not desirable for customers, given the postulation that prices should be known ahead of time [FD98, KSWS98]. Instead, we believe that a certain fraction of the overall network capacity is assigned to best-effort traffic and priced according to a traditional method (flat-fee, etc.).

- In [WPS97], communication services and demand patterns are modelled by the notion of calls, i.e., call probability, call duration, static QoS, etc. While being applicable to ATM service classes, this model does not fit well with the IntServ framework. Instead, our model uses aggregated demand functions for each time period, which implicitly encompasses the above details and also covers dynamic QoS.

The core formula which shows the total revenue that is to be optimized can then be specified using (6), (7) and (8) and looks as follows (roughly using the notation of [WPS97]):

$$\int_0^{Tb} \left\{ \sum_{X = T, G, R} \gamma_X(p_X, t) \times p_X \right\} dt - K(C_{Tb}) \qquad (12)$$

under constraints

$$\gamma_T(p_T, t) \le C_{Tb} \qquad (13)$$

$$\gamma_G(p_G, t) \le C_{Tb} - \gamma_T(p_T, t) \qquad (14)$$

$$\gamma_R(p_R, t) \le C_{Tb} - \gamma_T(p_T, t) \qquad (15)$$

Variables used:

p_X: price for each unit of virtual resource q_X (equal to a, b or c from (1), resp.)

$\gamma_X(p_X,t)$: aggregated demand for q_X at time t, when price is p_X

Tb: duration of business cycle

C_{Tb}: total available service rate (reservable bandwidth)

$K(C_{Tb})$: amortization of capital investment over one cycle

Constraints (13), (14) and (15) denote the fact that the amount of service rate reserved as token rate cannot be re-used, whereas service rate used as clearing rate can be used simultaneously as residual rate.

Comparing (12) with the corresponding formula in [WPS97] shows that using virtual resource parameters and considering only aggregated demand significantly reduces the mathematical complexity. While being subject of ongoing work, it is our assumption that in such a way, the problem of optimal pricing might be analytically tractable. We are convinced that our approach is very useful to apply theoretic results in a real environment.

5.2 Full-Cost Calculation

In [FD98] it has been pointed out that there might be situations in which cost calculation has to be based on full costs, instead of opportunity costs. For example, if the communication market is regulated by a government agency, a network provider must prove real costs as the basis for its price calculation. In such a situation, a slightly modified cost model can be applied. Instead of estimating the revenue, full costs are assigned to a time period and divided among the service classes. The aggregated future demand is estimated for that time period as well, potentially based on past experience.

To operate economically, all costs have to be covered by the aggregated revenue, therefore the same methods can be applied as with opportunity costs, except in equations (6), (7) and (8) the left side is replaced by an appropriate fraction of the total cost. This procedure is highly useful, because it simplifies the task of estimating demand. This is due to the fact that aggregated demand can easier be estimated than exact demand on a small time-scale. Additionally, costs are better comparable between multiple service classes when using uniform cost coefficients in cost functions as in (9), (10) and (11).

5.3 Cost Allocation for Multicast Communication

The IntServ framework extensively builds upon usage of multicast communication. A thorough study of allocating costs among members of a multicast group is presented in [HSE97]. Cost allocation is described by splitting each link's costs among a defined subset of group members. Definition of the subset determines the allocation strategy. Of course, the sum of each cost fraction must equal the total costs for a link. Realizing such an approach becomes much simpler, if costs can be expressed as a linear function of resource parameters, especially if charges are shared among receivers with heterogeneous QoS requirements. The cost functions (9), (10) and (11) fulfil this requirement and therefore, simplify cost allocation for multicast communication.

6 Application to Charging Mechanisms

In [KSWS98], an approach to exchange charging information between RSVP routers is presented. The problem of appropriately representing prices was left open for further study. Using the methods presented in Section 4, we can establish a concise notion for prices which fits with the protocol-oriented approach of [KSWS98]. Although in [KSWS98] it was assumed that price representation probably depends on the service class, we can now formulate a single price function representing all service classes considered in this paper:

```
price :=      price for q_T
              price for q_C
              price for q_R
              max buffer-rate ratio
              other charge components
```

Using this notion, all necessary QoS-dependent price information is transmitted. There might be other charge components, for example a flow setup fee. This is represented by the generic field `<other charge components>`. The field `<max buffer-rate ratio>` represents the limited buffer space of each router. As discussed in Section 4.1, routers can be equipped, such that buffer space should never really be a scarce resource. Prices can be accumulated at each hop and because the price function is linear, upstream charges can easily be split at multicast branches (see also Section 5.3).

Note that even when the charge coefficients for each router are largely stable, it is usually necessary to transmit price information with each PATH message (see [KSWS98] for details). According to the *Edge Pricing* paradigm [SCEH96], the price function expresses the total accumulated charges from the sender to the respective next hop. Therefore, accumulated price functions for different flows using different paths are very likely to differ.

It is clear that an indirect price representation like this adds additional complexity to the end systems, in that this price representation has to be translated into a user-friendly format. However, translation of QoS parameters has to take place for IntServ requests anyway and it is a common design paradigm in the Internet to push intelligence towards the end systems while letting the network technology be as simple as possible. Therefore, wo do not believe this slight additional complexity to be a problem.

7 Summary and Future Work

In this paper, we discussed charging and resource aspects related to cost and price calculation for IntServ communication services. We presented a method called *virtual resource mapping*, which can be used to apply well-known economic principles to IntServ cost calculation. We showed how existing theoretic results related to price and cost calculation can be used with virtual resource mapping and also how charging mechanisms can employ this method.

We are currently in the process of implementing the charging mechanisms introduced in [KSWS98], which are embedded in RSVP. With the forthcoming implementation we will be able to run extensive simulations of charging procedures and pricing algorithms incorporating the ideas presented in this paper.

References

[Asa98] Manjari Asawa. Measuring and Analyzing Service Levels: A Scalable Passive Approach. In *Proceedings of 6th IEEE/IFIP International Workshop on Quality of Service, Napa, CA, USA*, pages 3–12. IEEE/IFIP, May 18–20 1998.

[BCS94] Robert Braden, David Clark, and Scott Shenker. RFC 1633 - Integrated Services in the Internet Architecture: an Overview. Informational RFC, June 1994.

[BZB+97] Robert Braden, Lixia Zhang, Steve Berson, Shai Herzog, and Sugih Jamin. RFC 2205 - Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. Proposed Standard, September 1997.

[CSKW98] C. Courcoubetis, V. A. Siris, F.P. Kelly, and R. Weber. A Study of Simple Usage-based Charging Schemes for Broadband Networks. In *Proceedings of Broadband Communications '98, Stuttgart, Germany*, April 1998.

[CSZ98] Georg Carle, Michael Smirnov, and Tanja Zseby. Charging and Accounting Architectures for IP Multicast Integrated Services over ATM. In *Proceedings of 4th International Symposium on Interworking, Ottawa, Canada*, July 1998.

[DKS89] Alan Demers, Srinivasan Keshav, and Scott Shenker. Analysis and Simulation of a Fair Queueing Algorithm. In *Proceedings of ACM SIGCOMM '89, Austin, TX, USA*, pages 1–12, September 1989.

[DVR98] Konstantinos Dovrolis, Maruthy Prasad Vedam, and Parameswaram Ramanathan. The Selection of the Token Bucket Parameters in the IETF Guaranteed Service Class. Technical report, University of Wisconsin-Madison, Madison, WI 53706-1691, USA, July 1998.

[FD98] Domenico Ferrari and Luca Delgrossi. Charging for QoS. In *Proceedings of 6th IEEE/IFIP International Workshop on Quality of Service, Napa, CA, USA*, pages vii–xiii. IEEE/IFIP, May 18–20 1998. Invited paper.

[FO98] Peter C. Fishburn and Andrew M. Odlyzko. Dynamic Behavior of Differential Pricing and Quality of Service Options for the Internet. In *Proceedings of First Intern. Conf. on Information and Computation Economies (ICE-98)*, 1998.

[FSVP98] George Fankhauser, Burkhard Stiller, Christoph Vögtli, and Bernhard Plattner. Reservation-based Charging in an Integrated Services Network. In *4th INFORMS Telecommunications Conference, Boca Raton, Florida, USA*, March 1998.

[Gal97] John Gallant. Are you ready for those management challenges? *Network World*, August 4 1997.

[GGPR96] Leonadis Georgiadis, Roch Guerin, V. Peris, and R. Rajan. Efficient Support of Delay and Rate Guarantees in an Internet. In *Proceedings of ACM SIGCOMM'96*, pages 106–116, August 1996.

[GSW95] Alok Gupta, Dale O. Stahl, and Andrew B. Whinston. A Stochastic Equilibrium Model of Internet Pricing. In *Seventh World Congress of the Econometrica Society, Tokyo, Japan*, August 1995.

[HSE97] Shai Herzog, Scott Shenker, and Deborah Estrin. Sharing the "Cost" of Multicast Trees: An Axiomatic Analysis. *IEEE/ACM Transactions on Networking*, 5(6):847–860, December 1997.

[KSWS98] Martin Karsten, Jens Schmitt, Lars Wolf, and Ralf Steinmetz. An Embedded Charging Approach for RSVP. In *Proceedings of 6th IEEE/IFIP International Workshop on Quality of Service, Napa, CA, USA*, pages 91–100. IEEE/IFIP, May 18–20 1998.

[KVA98] Yannis A. Korillis, Theodora A. Varvarigou, and Sudhir R. Ahuja. Incentive-Compatible Pricing Strategies in Noncooperative Networks. In *Proceedings of the 17th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'98)*, March 1998.

[Lei98] Brett A. Leida. Cost Model of Internet Service Providers: Implications for Internet Telephony and Yield Management. Master's thesis, Massachusettes Institute of Technology, Cambridge, MA, USA, February 1998.

[MMV95] Jeffrey K. MacKie-Mason and Hal R. Varian. Pricing the Internet. In Brian Kahin and James Keller, editors, *Public Access to the Internet*, pages 269–314. MIT Press, Cambridge, 1995.

[MMV97] Jeffrey K. MacKie-Mason and Hal R. Varian. Economic FAQs About the Internet. In Joseph Bailey and Lee McKnight, editors, *Internet Economics*. MIT Press, Cambridge, 1997.

[PG93] Abhay K. Parekh and Robert G. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, June 1993.

[PG94] Abhay K. Parekh and Robert G. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Multiple Node Case. *IEEE/ACM Transactions on Networking*, 2(2):137–150, April 1994.

[SCEH96] Scott Shenker, David Clark, Deborah Estrin, and Shai Herzog. Pricing in Computer Networks: Reshaping the Research Agenda. *ACM Computer Communication Review*, 26(2):19–43, April 1996.

[SFY95] Jakka Sairamesh, Donald F. Ferguson, and Yechiam Yemini. An Approach to Pricing, Optimal Allocation and Quality of Service Provisioning in High-Speed Packet Networks. In *Proceedings of the 14th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'95)*, pages 1111–1119. IEEE Computer Society Press, June 1995.

[SPG97] Scott Shenker, Craig Partridge, and Roch Guerin. RFC 2212 - Specification of Guaranteed Service. Proposed Standard, September 1997.

[WPS97] Qiong Wang, Jon M. Peha, and Marvin A. Sirbu. Optimal Pricing for Integrated-Services Networks with Guaranteed Quality of Service. In Joseph Bailey and Lee McKnight, editors, *Internet Economics*. MIT Press, 1997. Available at http://www.ece.cmu.edu/%7epeha.

[Wro97] John Wroclawski. RFC 2211 - Specification of the Controlled-Load Network Element Service. Proposed Standard, September 1997.

Appendix

The following example calculations are provided to enable a real-world point of view on the relation of buffer and service rate requirements for IntServ data flows. Both examples are calculated using the formulas given in RFC 2212 [SPG97], although example 2 also roughly applies to a Controlled Load or Guaranteed Rate scenario. Usage of C and D error terms from Guaranteed service slightly increases the buffer requirements as can be seen in the appropriate formulas. Note that previous routers along the flow's path usually have smaller C and D values to cope with, hence, the buffer requirements would be smaller, as well.

Example 1 Conferencing using MPEG-1 sized video encoding

We consider a videostream with its typical 1.5 MBit/s average data rate. The burst rate is set to 3 times the average rate and the burst duration is set to 1.5 seconds. The required end-to-end delay is set to 300 milliseconds, such that humans will not explicitly notify any latency. This should cover a usual videoconferencing scenario.

traffic description (TSpec):

p	4.5 Mbit/s	b	4.5 Mbit	r	1.5 Mbit/s
M	1500 bytes	m	100 bytes		

error terms:

C_{tot}	15000 bytes	D_{tot}	50 msec

requested bound on end-to-end delay: 300 msec

results:

required service rate: 3931264 bit/s ≈ 4 Mbit/s

required buffer: 147422 bytes

buffer-to-rate ratio: approx. 0.0375 $\frac{\text{MB}}{\text{Mbit/s}}$.

Example 2 Playback of large and bursty videostream

In this example, we consider the transmission of a large and bursty videostream, for example for a high-quality video-on-demand application. We assume that delay does not matter, which in reality would require an end system to provide a large playout buffer. However, combination of a large burst size with a low service rate imposes the highest requirements on buffer space for routers, therefore this scenario was chosen.

traffic description (TSpec):

p	20 Mbit/s	b	40 Mbit	r	5 Mbit/s
M	1500 bytes	m	100 bytes		

error terms:

C_{tot}	15000 bytes	D_{tot}	50 msec

requested service rate: 5 Mbit/s

results:

resulting bound on end-to-end delay: 8074 msec

required buffer: 5046250 bytes ≈ 5 MB

buffer-to-rate ratio: approx. 1 $\frac{\text{MB}}{\text{Mbit/s}}$.