[KSW Ste98] *Martin Karsten, Jens Schmitt, Lars Wolf, Ralf Steinmetz*; **An Embedded Charging Approach for RSVP;** Sixth IEEE/IFIP International Workshop on Quality of Service (IWQoS'98), Napa, CA, USA, 18.-20.05.98, ISBN 0-7803-4482-0, 10 S.

An Embedded Charging Approach for RSVP

Martin Karsten[1] , Jens Schmitt[1] , Lars Wolf[1] , and Ralf Steinmetz[1,2]
[1] Darmstadt University of Technology *
[2] German National Research Center for Information Technology, GMD IPSI

Abstract Charging mechanisms 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. In this paper, we describe a charging model that can be embedded in the RSVP architecture. Our model is open and flexible in that it imposes little or no restrictions to the pricing policy of network providers or the usage behaviour of end-users. At the same time, it provides mechanisms to enable fine-grained charging of network communication. After a user-centric identification of requirements for charging mechanisms, a formal framework is presented to model the prices and payments. We present protocol elements and implementation rationale to realize our charging model. Furthermore, we identify potential problems that are inherent to RSVP with regards to precise charging and point out future research issues towards a realistic charging architecture.

Keywords QoS, Charging, RSVP, Policy Control.

1 Introduction

Over the last years, the Internet has evolved from a closed community network into a public, commercial communication system, used not only by researchers and academic institutions, but also for private and business communication, marketing, commerce, etc. Currently, the Internet provides only a single class of unreliable service, in that each packet is treated independently and equally. This service model is referred to as *best-effort service*.

In the future, the Internet technology is expected to enable the creation of an *integrated services network* that eventually replaces other existing networks (telephony, cable-TV) to a large extent by offering a wide variety of services based on a single network infrastructure. However, we expect network resources to be scarce for quite a long time, opposite to the opinion that further advances of networking technology can enable the creation of a well-dimensioned network without resource bottlenecks. First, experience shows that any increase in the power of networking (or e.g. computer hardware) resources is quickly soaked up by new resource-demanding applications. Second, a network provider can quickly run into problems, if there is no feedback mechanism and usage patterns change significantly. As an adequate example there are reports about congestion and resulting problems on local telephone networks caused by residential Internet subscribers in North America [Mor98].

To support real-time transmission of continuous-media streams over scarce network resources, it must be discriminated between different service classes and different data flows, thus, their respective quality-of-service (QoS) requirements must be enforced by resource reservation. Currently, RSVP [BZB+97] is expected to provide the means to signal resource reservation requests from end systems to the network and between intermediate systems within the network.

Due to concerns about the scalability of RSVP, a new approach called Differentiated Services (DiffServ) has been proposed in the IETF [NB98]. In the DiffServ architecture, it is planned to define standard forwarding semantics for certain types of packets, which are marked by hosts and/or edge routers. Concatenation of these forwarding semantics leads to certain traffic classes. A Service Level Agreement (SLA) describes a traffic profile between one or many network participants and establishes a pipe with certain QoS attributes along a data path (or parts hereof). Here charging largely depends on the dynamics of SLAs. If an SLA is rather static, charging can be done off-line, otherwise a signalling protocol is needed, which might turn out to be (similar to) RSVP. However, it is our firm belief that an integrated services network eventually needs precise and flow-specific resource reservation.

It is obvious that if network traffic can be protected by individual resource reservation, some negative feedback is needed to prevent users from arbitrarily allocating resources. On the other hand, a market and competition mechanism is needed to provide users with the best and most inexpensive level of service, while creating incentives for network providers to supply more resources when there is sufficient demand. Therefore, charging mechanisms are needed to compensate for the allocation of scarce resources.

In this paper, we describe a charging approach for RSVP. We adhere to the principle of separating mechanism and strategy in that we try to impose as little pricing strategy as possible. Information about prices and other charging details are embedded in POLICY_DATA objects, which are part of various RSVP messages. The primary goal is a charging approach which is fine-grained,

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

convenient, comprehensible and secure for users in their relation to network providers. Therefore, we devise our approach to achieve similar charging characteristics as known from telephony. Our underlying assumption is that any less customer-friendly charging scheme will prohibit the general acceptance of an integrated services network. Furthermore, our approach allows for partial deployment in the global Internet environment.

Most research work that has been carried out about communication charging either theoretically approaches the problem of setting optimal prices and/or remains rather vague when it comes to actually calculating total prices and describing accounting. Theoretical analysis and observations from simulations usually have too many restricting assumptions to be applicable to real networks. While this research work is very important to gather insight in the fundamentals of pricing theory, its practical relevance is at least questionable. [SCEH96] were the first to clearly point out this insufficiency, which is particularly important with respect to the existence of multiple *independent* network providers and a *heterogeneous* network structure, both of which have to be considered when designing charging mechanisms. Furthermore, it is important to realize the fundamental principle that setting a price for a service is under the authority of the service provider, except where certain limited market regulations apply. Given the general assumption that a competitive market creates the best possible service value for all customers, this principle must not be denied for charging of network communications. The design of our charging approach is driven by a strict "real-world" attitude, in that we try to present and discuss mechanisms that are as flexible as possible, while being specific enough not to leave out important details.

The rest of the paper is organized as follows. Section 2 formulates general principles for charging communication services. In Section 3 we describe a formal model for the flow of price information and payments and how it is utilized in the context of RSVP. A critical revision of the charging approach is done in Section 4. Finally, Section 5 summarizes our results and gives an outlook to future research issues.

2 Goals and Expectations for Charging of Communication Services

Some fundamental assumptions about the relationship between market participants have to be reviewed when the Internet is viewed as a commercial communication network where users are charged according to their resource consumption. These assumptions are mainly driven by the individual market participant's point of view, opposite to previous approaches that try to find optimal solutions for network charging, e.g., [SFY95,WPS97].

- Each participant is independent and individually seeks to minimize its costs while maximizing its profit. This assumption fundamentally contradicts the request for a global optimal price function.
- Participants do not necessarily trust each other, not only with regard to authentication, but also in terms of correct information.
- Participants are used to a high level of legal security.
- Customers are used to a high level of service and customer protection.
- Communication prices are set independently by each network provider, *but* the price for a service likely depends on the costs for sub-services that are needed from other providers.

2.1 User Requirements and Expectations

Given these assumptions, a number of user expectations for charging of communication services can be deduced and the requirements derived from them must be addressed by charging mechanisms. This assembly of expectations and requirements is heavily influenced by observation of today's telephone market.

Predictability of Charges. Users want to be able to predict the costs of using a particular application, which include the expenditures for the communication services induced by this application. Therefore, an exact a priori specification of communication charges would be desirable. However, if this requirement cannot be fulfilled, a set of weaker demands can be sufficient. First, a user should be able to roughly estimate its charges. Such an estimation does not need to be exact but should give at least a rough feeling to the user – similar like the knowledge that an international phone call of a few minutes duration costs several dollars and not just a few cents. Second, a worst-case price should be announced to the users. Finally, it must be prohibited that a user is charged a higher price than previously announced, without giving her explicit approval.

Stability of Service. When a particular service with a certain quality has been agreed upon by the user and the provider, it must be ensured that the service indeed is delivered to the user. Hence, an exact definition of "quality assurance is met" is needed. On the other hand, users must be able to estimate the impact of such quality goals on their applications, hence the definition must not be too complex. For example, multiple users start a video conference application, thus they likely request a communication service with a specified bandwidth and delay. If the provider assures to deliver this service, the users expect no quality degradation and a very low probability of service disruption during the conference. In case of quality degradation or service disruption, an appropriate refund mechanism must be applied, which largely depends on the type of application, and hence, should be negotiated during set up of the communication service.

Transparency and Accuracy of Charging. To find out how much is spent for which application and what are the reasons for this, users need the ability to determine the costs of a particular session, e.g., if an application uses several flows, the costs for each of these should be stated explicitly. Furthermore, for some users it might also be of interest to see where inside of the network the major charges are caused. This may give them information to switch to a different provider in future. Detailed per-session information about charges can also be used to decide whether a certain service and its quality offer good value for the price. Since not all users are interested in such details, each user must be able to decide how much information should be given.

Flexibility. When information is transmitted from a sender to one or several receivers, the flow of value associated with this information can be (1) in the same direction as that of the data flow, (2) in the opposite direction, or (3) a mixture of both because both sides benefit from the information exchange. For example, in the first case, the sender transmits a product advertisement, in the second case, the receiver retrieves a movie for playback, and in the third case both sides hold a project meeting via a video-conference system. To support these different scenarios, a charging architecture must provide flexible mechanisms to allow the participants in a communication session to specify their willingness to pay for the charges in a variety of manners. Senders must be able to state that they accept to pay for some percent of the overall communication costs or up to a specified total amount. Similarly, receivers may state what amount of costs they will cover. Additionally, charging mechanisms must allow to flexibly distribute communication charges among members of a multicast group. A number of cost allocation strategies can be found in [HSE97].

Fraud Protection and Legal Security. One of the most important issues demanded by participants is protection against fraud, i.e., that they do not have to pay for costs they have not incurred and that no one can misuse the system. The fear of users is that a provider may cheat or that other users may use their identity or derogate from them in any other way. Providers want to be sure that users indeed pay for the used service. A prerequisite against fraud is technical security, such that users cannot damage, misuse or intrude the provider's communication systems. Finally, legal security denotes the demand that in case of a failure, there is enough information to determine responsibility for it.

Technical Feasibility. The charging approach and its mechanisms must be realizable and usable with low effort. Otherwise, if it becomes too complex, the costs for the charging mechanisms might be higher than their gains. The added overhead for communication due to additional information transmitted between senders, network nodes, and receivers, and also for processing and storage purposes especially in network nodes, e.g., to keep and manipulate charging information, must be as low as possible.

Convenience. Charging components should not make the use of communication services much more difficult. The charging mechanisms themselves as well as the final bill based on the information gathered by the charging system must be convenient for the users. Hence, it must be possible for users to define "standard charging behaviour" for their applications so that they are not bothered with details during the start up of an often used application. On the other hand, they should be able to change such a description easily to have control over their expenditures. Furthermore, most users want to have as few separate bills as possible, i.e., have contracts and according business procedures with only one provider.

2.2 Framework for the Charging Approach

A fundamental aspect of our charging approach is the use of the *Edge Pricing* [SCEH96] approach. Corresponding to this paradigm, a user is charged only by the first network provider along the data path. This charge includes all expenses that subsequently might have to be paid by the provider when data is forwarded to another provider. While in principle a market participant may have business relations to multiple other participants, every single service instantiation is requested from and charged by exactly one peer participant. Edge Pricing is not necessarily needed to accomplish the requirements mentioned above, but it is an appealing paradigm that helps meeting demands like transparency, flexibility, convenience and legal security. Edge Pricing reduces the problem of multi-lateral contracts to a sequence of bilateral contracts and therefore hides much of the complexity which is introduced by the existence of multiple service providers and heterogeneous networks in the communication path.

3 A Charging Approach for RSVP

The RSVP specification already incorporates hooks for policy-related actions, for example the exchange of POLICY_DATA objects. In this section we explain a charging approach for data flows for which a certain QoS is being reserved using RSVP. The mechanisms cover both unicast and multicast transmission, as well as sharing of transmission costs between senders and receivers. After giving a general introduction to the approach, we present a formal framework to model the flow of pricing information and payments. Finally, we describe the proposed definition of the policy elements, their semantics and the mechanisms how to use them. We only give rough definitions for a variety of reasons. The exact definition of a pricing function depends on the service class that is actually chosen to transmit data. Furthermore, the definition and semantics of protocol elements is largely a matter of local agreement between the operators of two adjacent RSVP-capable hops. Last not least, we believe in the need for further research and discussion to fully understand the impact and dynamics of a fine-grained charging approach like this one.

While we describe the charging mechanisms in terms of applying them at each hop on the data path, it should be easy to see that this is not a necessary requirement. Therefore, it is possible to partially deploy our approach in the Internet and even inter-operate with other charging mechanisms. This is an immediate implication of adhering to the Edge Pricing paradigm.

3.1 Basic approach

We assume a general layout of the underlying network and a general model of an RSVP session, such that multiple providers and end-users may be involved. This is schematically shown in Figure 1. At this point, we make no

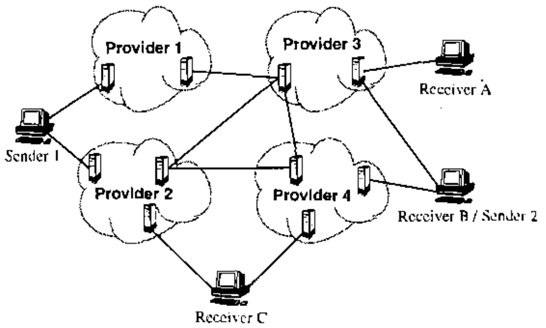

Figure 1: RSVP Session in Multi-Provider Network

restrictions about the complexity of an RSVP session.

In RSVP, the ability to reserve resources is announced by sending PATH messages along a data path. Reservations are initiated by receivers of a data flow by responding with RESV messages. Therefore, a straightforward approach is to collect reservation charges from receivers. However, when a RESV message travels upstream, each RSVP-capable router initiates a resource reservation at the previous hop router by sending a (possibly modified) RESV message for this session. It is likely that a charge applies for this reservation as well. According to the edge pricing paradigm, the end user is charged by its direct network provider and each provider is charged by the next provider upstream. Thus, the final reservation price is basically the sum of the prices of all network providers along the data path.

We define the necessary protocol information according to the general policy extension proposed in [Her96,Her97], but the charging mechanisms can as well be realized using a different general framework. According to [Her96,Her97], part of a POLICY_DATA object is a *policy element list*, which is not further defined in the referred proposal. Therefore, we define new *policy elements* that are used for charging. Further, [Her96,Her97] specify that dedicated *policy handlers* within a *Local Policy Module (LPM)* are responsible for the handling of policy elements. For the purpose of this discussion, we denominate the charging-related handlers *charging handlers* and collectively *Local Charging Module (LCM)*. In terms of the proposed general policy architecture, the LCM is part of the LPM and the charging handlers are specific policy handlers.

At this point, it is important to notice that multiple types of LCM are possible: the ones that are in the edge routers of a provider's network and the ones that are in routers inside the administrative domain of a provider. The former LCM needs much more functionality than the latter, although one could argue that the latter could have a similar functionality in order to allow for some internal accounting, however the external accounting functions will have more challenges to cope with due to phenomena like fraud which arise in a competitive environment.

The basic idea of our charging approach is to construct a *Downstream Charging Policy Element (DCPE)* data structure that is sent downstream within the POLICY_DATA object of PATH messages. The fields in this structure are used to collect the providers' prices as well as other charging related information. Intermediate routers build soft state from this information within their LCM, corresponding to PATH and RESV state of RSVP. Further, the price information is updated by the LCM according to the provider's pricing policy. Upon arrival of a PATH message at the receiver's end system, the total charge has been manifested and the receiver decides whether it is willing to pay this charge to reserve resources. If yes, it issues a RESV message containing an appropriate *Upstream Charging Policy Element (UCPE)* within the POLICY_DATA object. The same mechanism is applied at intermediate nodes, such that in general the arrival of a RESV message indicates the downstream hop's consent to be charged for a reservation.

In general, a network provider is interested in providing the service, i.e., establishing the reservation, in the first place. Therefore, we assume that an RSVP hop sets the lowest reasonable price in a DCPE to attract potential customers. Additionally, customers might be able to select from multiple network providers as for example receivers B and C in Figure 1, creating a competitive environment. On the other hand, if there is a highly requested link in the network that allows a provider to set arbitrarily high prices, market forces will bring up competitors to provide an equivalent service.

3.2 A Formal Model of Charges and Payments

In order to explain how charges and payments are calculated, we give a formal definition of the necessary parameters and show by solving an appropriate equation that all payments lead to exact revenue of the calculated local price for each RSVP hop. For the purpose of explanation, we initially restrict the model to only one sender. An important issue is how to represent prices and payments. In our model a price is a *price per resource unit.* In this context, the term *resource unit* is largely dependent on the representation of the communication service class, e.g., if the service class offers the parameters *bandwidth* and *delay*, the price depends on these parameters. Additionally, the total price for a communication session depends on the duration of this session. In reality, a payment can be represented, for example, as a direct exchange of virtual money or a credit or debit to an account.

In the following we present a formal model of a network, charges and payments, as well as their allocation to a sender and multiple receivers.

Let $i = 0,...,n,n+1,...n+m$ be a number of nodes in a multicast session, where

$i = 0$ denotes the sender,
$i = 1,...,n$ denote an intermediate router, and
$i = n+1,...,n+m$ denote a receiver

We define a *multicast function* $m(j)$ that denotes the previous hop for a node j.

$m : \{1,...,n+m\} \rightarrow \{0,...,n\}$ with these characteristics:

$m(j) = 0$ for at least one $j \in \{1,....,n\}$
$m(i) \neq i$ for all $i \in \{1,....,n\}$
$m(i) = j \Rightarrow m(j) \neq i$

To make the charging procedure as transparent as possible for the sender and all receivers, the model is based on a *total charge*, which is eventually known to all end systems. Let C_i denote the total charge that has to be paid (by whoever) to connect hop i to the multicast tree. In order to recover this amount, a node splits it (according to a local policy) into multiple fractions $c_{i,j}$ for each outgoing interface where a reservation is established. A local price $L_{i,j}$ depending on the providers local price scheme is added.

Let $c_{i,j}$ denote a fraction of C_i for $m(j) = i$, with

$$\sum_{j,\, m(j) = i} c_{m(j),j} = C_i$$

Let $L_{i,j}$ denote the local price for a reservation on the outgoing interface to j when $m(j) = i$.

The total charges for a hop can be calculated as follows:

$$C_j = c_{m(j),j} + L_{m(j),j}$$

Between two adjacent hops, a charge is paid in the upstream direction. This payment is eventually recovered from the receiver end systems. The paid charge consists of a fraction of the charge until the current hop and the local price at the current hop. Let $RP_{i,j}$ denote such a receiver payment from a downstream hop i to an upstream hop j:

Let r be the *fraction* the sender is willing to pay, so the receiver has to pay a fraction of 1-r.

$$RP_{i,j} = (c_{m(i),i} + L_{m(i),i}) \times (1\text{-}r) = C_i \times (1\text{-}r) \text{ for } m(i) = j$$

Additionally, there are downstream payments that are eventually recovered from the sender. The charge consists of the previously paid downstream payments and the sender fraction of the local price at the current hop. Let $SP_{j,i}$ denote a sender payment from an upstream hop j to a downstream hop i:

$$SP_{j,i} = \sum_{k,\, m(k) = i} SP_{i,k} + \sum_{k,\, m(k) = i} L_{m(k),k} \times r$$

Finally, let E_j denote the earnings at node j. We define them as the difference between the incoming and outgoing payments and show that this is equal to the sum of local prices:

$$E_j = \sum_{k,\, m(k) = j} RP_{k,j} + SP_{i,j} - \sum_{k,\, m(k) = j} SP_{j,k} - RP_{j,i}$$

for $m(j) = i$

It follows that:

$$E_j = \sum_{k,\, m(k) = j} (c_{m(k),k} + L_{m(k),k}) \times (1 - r)$$
$$+ \sum_{k,\, m(k) = j} SP_{j,k} + \sum_{k,\, m(k) = j} L_{m(k),k} \times r$$
$$- \sum_{k,\, m(k) = j} SP_{j,k} - (C_j \times (1 - r))$$

$$= \sum_{k,\, m(k) = j} c_{m(k),k} \times (1 - r) + \sum_{k,\, m(k) = j} L_{m(k),k} \times (1 - r)$$
$$+ \sum_{k,\, m(k) = j} L_{m(k),k} \times r - (C_j \times (1 - r))$$

$$= (C_j \times (1 - r)) + \sum_{k,\, m(k) = j} L_{m(k),k} - (C_j \times (1 - r))$$

$$= \sum_{k,\, m(k) = j} L_{m(k),k}$$

If receivers specify shared reservation styles that apply to at least one common sender, they are merged on shared links. In that case, each sender's fraction must not directly be applied to the single charge on a shared link, otherwise the distribution of payments does not come out correctly. If for example two senders independently specify to cover half of the charge, the use of shared reservation style would cause them to effectively pay for the total cost of a shared link, whereas a receiver might get away for free. We use the following definition to formally handle this case. However, in order not to let our model become too complex, we do not consider this case in the rest of this section.

Let $L_{s,i,j}$ denote the local price for a fixed filter reservation regarding sender *s*.
The price $FFP_{s,i,j}$ for a fixed filter reservations can then be expressed as:

$$FFP_{s,i,j} = (c_{i,j} + L_{s,i,j}) \times (1 - r)$$

Let $SFP_{i,j} = \max_{s \in SF}(FFP_{s,i,j})$ denote the price for a single shared reservation style from hop *i* to hop *j*.
Let r_s denote the charging fraction for sender *s*.
When SF denotes the set of senders merged by a shared reservation style, the sender payment can be expressed as:

$$SP_{s,j,i} =$$

$$\sum_{k,\, m(k) = i} SP_{i,k} + \sum_{k,\, m(k) = i} SFP_{i,k} \times r_s \times \frac{FFP_{s,i,k}}{\sum_{s,\, s \in SF} FFP_{s,i,k}}$$

The definition of $RP_{i,j}$ has to be modified accordingly.

In the rest of the paper, we use the following convention when definitions of this model are referred to:
- i denotes the previous hop
- j denotes the current hop
- k denotes any next hop

3.3 State Information within the LCM

For the purpose of explaining our approach, we describe what state information is likely to be stored in an LCM. Of course, this does not prohibit implementations to internally differ from these suggestions. The LCM keeps state information for each pair (Session, Sender). We call this state *downstream charging state* and define it as follows:

```
downstream
charging state ::=   <total charge upstream>,
                     <max total charge upstream>,
                     <sender fraction>,
                     <sender account>,
                     <limit per receiver>,
                     <limit per hop>,
                     <max number of hops>
```

The field <total charge upstream> holds the charging amount C_j that is announced from the previous hop. This information might change with subsequent DCPEs, therefore a maximum number is stored in <max total charge upstream>. The sender's willingness to pay for communication is expressed by the field <sender fraction>. It stores the fraction *r* that the sender is willing to cover. This value applies at each single hop and therefore, to the complete data path as well. In analogy to the definition of $RP_{j,i}$, <total charge upstream> and <sender fraction> together represent a claim from the previous hop to the current hop in case the current hop requests a reservation. The other four fields are used to buffer further information about the sender's account and maximum share of costs. Their usage is explained in Section 3.4.

Additionally, *upstream charging state* is stored for each triple (Session, Next hop, FilterSpec) when appropriate UCPEs within RESV messages are admitted:

```
upstream
charging state ::=   <local price>,
                     <payment downstream>,
                     <total charge downstream>
```

In <local price>, the RSVP hop stores the local price $L_{j,k}$ that currently applies for a reservation on the appropriate outgoing interface. If a sender agrees on paying a share of the reservation charge, a debit is accumulated upstream hop by hop, corresponding to the definition of $SP_{j,k}$. The current hop's payments are buffered in <payment downstream>. For transparency reasons, the total reservation charge is calculated when RESV messages are sent upstream. The total charge that is reported from a next hop is buffered in <total charge downstream>. This value together with the sum of all local prices is delivered to the previous hop.

The complexity of state information for downstream and upstream charging state corresponds to RSVP PATH and RESV state, respectively. Therefore, handling of state information can be expected to have scalability characteristics similar to RSVP itself. Actually, an implementation might choose to store LCM state together with the respective RSVP state, however, for the purpose of our description we rather keep them logically separated.

3.4 Downstream Charging Policy Element

The *Downstream Charging Policy Element (DCPE)* is defined as follows:

```
DCPE ::=             <total charge>,
                     <max total charge>,
                     <duration of price validity>,
                     [ <sender's share> ]

<sender's share> ::= <sender fraction>,
                     <sender account>,
                     [ <limit per receiver> ],
                     [ <limit per hop> ],
                     [ <max number of hops> ]
```

The incoming <total charge> field contains the accumulated total charge C_j up to the current node and is stored in the <total charge upstream> field of downstream charging state. When the set of next hops is determined for a PATH message, the local charges $L_{j,k}$ for a reservation on the corresponding outgoing interfaces are calculated depending on the providers local price policy and stored in the <local charge> field of upstream charging state. The sum of both values is used to create the <total charge> field of an outgoing DCPE.

The prices in a communication network are expected to change over time, depending on the calculations of network providers, both in the short term (due to congestion situations) and in the long term. The <duration of price validity> field indicates how long the upstream hop assumes the current price information to remain stable. This information is used to set the timeout for downstream charging state built from a DCPE. It is important to notice that a previous hop can hardly be held liable for this price information. Merging of reservations definitely influences the price calculation of a network provider. As an example, if a new receiver joins or leaves a multicast group, this might lead to the creation or deletion of a reservation on an outgoing interface, which in turn changes the cost-recovering price on the other interfaces. This might change the provider's price calculation, before an appropriate RESV message is received. However, even if providers could be forced to charge the announced price, an RSVP router might be implemented to simulate an admission control failure in such a case. It is mainly due to the receiver-initiated reservation model of RSVP that causes this possibility. For these reasons, the worst-case charge, i.e. basically the unicast charge along the path, is accumulated, as well. This is done by sending an appropriately filled <max total charge> field downstream. The information from an incoming DCPE is stored in <max total charge upstream> of downstream charging state. Added up with the highest possible local price $max(L_{j,k})$, this value is set in <max total charge> of outgoing DCPEs.

A sender can indicate its consent to cover a fraction of the total transmission charge. The <sender fraction> field (corresponding to <sender fraction> in downstream charging state) allows the sender to specify the fraction of costs it accepts to pay. Account information from the sender is stored in <account>. In order to protect the sender from arbitrarily high costs, it is necessary to restrict the maximum charging amount independently of any underlying restriction in distribution of data. A first approach would allow a sender to specify a maximum charging amount. However, there are obstacles to this procedure. Consider the case where a sender is interested in reaching a large user population with its data flow and sets a very high maximum amount. Each provider is independent in setting its prices, so if any provider had knowledge about a receiver that is connected directly to its network, it could set its price high enough to let the total sum be just below the maximum amount, but still be much higher than its normal price, hence, prohibit any other receiver to receive the data free of charge. The underlying problem here is the distributed and uncoordinated installation of reservations in RSVP without providing any global state. The solution is to have the sender give a more detailed specification of its interests. Rather than specifying the maximum charging amount, the sender specifies a maximum per receiver, i.e., per branch in the multicast tree (set in <limit per receiver>). Additionally, a sender can set an upper level of charges per hop (with <limit per hop>) and roughly restrict the geographic distribution of the sponsored flow (<max number of hops>). Together, the latter two fields can be used to restrict the sender's total charging amount to the product of both values. If the <max number of hops> field is set in an DCPE, it must be decremented before it is forwarded within an outgoing DCPE. The other two limitation fields must not be changed. It is important to notice that routers are automatically discouraged from changing those fields, because forgery can either be detected through end-to-end control or it is harmful for the forging router (being held responsible for it) in the first place. When a DCPE is processed, it has to be checked whether:

1) <max number of hops> is decreased to 0 or
2) the product of <sender fraction> and <total charge> exceeds <limit per receiver> or
3) the sum of all <local charge> exceeds <limit per hop>

If any of the above checks turns out to be true, no sender charging is done anymore and the <sender fraction> field should be set to 0 for any outgoing DCPE for this sender.

3.5 Upstream Charging Policy Element

The *Upstream Charging Policy Element (UCPE)* is defined as follows:

UCPE ::= <account>,
<payment>,
<total charge>,
<sender payment>

We assume that an authentication step is done by using the default RSVP integrity mechanism [Bak97] and potentially by using other policy mechanisms as described in [Her97]. Any information directly related to charging and accounting is given using the <account> field, for example the selection of a particular debit account, if desired. The <payment> field contains the total payment provided for this reservation. If <payment> covers the current receiver price (analogous to $RP_{j,k}$), the receiver price is charged to the given account. The validity of this payment is implicitly defined by the refresh timer value of the RESV message that carries an UCPE. As discussed in Section 3.4, there are timing problems related to using the *current* receiver price, which cannot precisely be fixed. Additionally, due the nature of RSVP, there is a time gap between initiating a reservation, i.e., sending the RESV message, and its actual installation in all intermediate systems up to the current node. To this end, it is not clear which participant is responsible for the charges that apply in the meantime. The situation becomes even more complex, if a reservation fails at a router and the previously installed reservation have to be torn down shortly after being installed.

In <total charge> the complete sum of charges that applied to this reservation is reported. This is stored in <total charge downstream> of upstream charging state. If any charge is passed on to the sender, a sender payment ($SP_{j,k}$) has been debited from the current hop's account (which is announced by the <sender account> field of a DCPE). This information is transmitted in the <sender payment> field.

When a RESV message eventually reaches the sender, the <payment> field carries the total charge paid by receivers for all reservations regarding this sender's data flow within a single RSVP session whereas the <sender payment> contains the fraction charged to the sender. Additionally, the <total charge> field carries the sum of all charges that applied.

3.6 Example Scenario

As an example, consider the scenario of a multi-party video conference. We explain the data flow from one sender to two receivers as shown in Figure 2. We assume that all

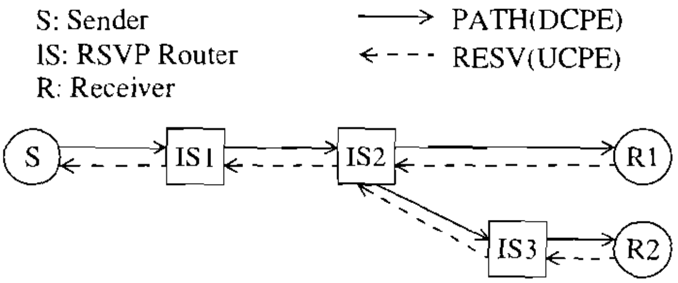

Figure 2: Example Scenario

RSVP routers belong to different network providers. Due to space limitations, this example cannot cover all the complexity that might be involved, but it should give a rough insight into the application of our charging approach.

Initially, the sender starts transmitting PATH messages describing the traffic characteristics of its video transmission. In this example, the sender is willing to pay 30% of the total transmission charges without limiting the hop count or the upper bound per receiver. In the initial DCPE, all required fields are set to zero. The field <sender fraction> carries the value 0.3 and <sender account> describes accounting information about the sender's account at IS1. When the DCPE reaches IS1, downstream charging state is created for it. There is only one outgoing interface concerned by this session, so the LCM of IS1 modifies the DCPE by storing the local price into <total charge> and <max total charge>. It fills <time of price validity> with an appropriate value. Then, a PATH message embedding the DCPE is sent to IS2. At IS2, two outgoing interfaces are concerned, so two modified copies of the DCPE are eventually sent downstream. For both DCPEs, the value of the <total charge> field is split, probably based on expected multicast characteristics, and an appropriate local charge is added. The maximum possible local price is added to <max total charge>, while <time of price validity> is overridden with a new value for each interface. Then, both DCPEs are sent downstream. The processing at IS3 is similar to that at IS1.

Eventually, the PATH messages reach R1 and R2, which in turn send RESV messages with appropriate UCPEs. We consider R2 first.

R2 calculates the necessary payment from the <total charge> and the <sender fraction> field. It sets the <payment> field in the UCPE accordingly. The value of <total charge> is transferred from the DCPE to the UCPE and the field <sender payment> is set to zero. Again, <account> is filled with appropriate account information. When the UCPE reaches IS3, the payment is accounted, <total charge> is increased by the local price and the appropriate payment to IS2 is written into <payment>. 30% (the content of <sender fraction>) of its local price is charged to IS2's account and written into <sender payment>. Similar processing happens while the PATH message travels upstream until it finally reaches the sender.

Let us assume that R1 also sets the exact payment information when requesting a reservation at IS2. Let us also consider that IS2 did a rather optimistic calculation of its local price on this interface and decides to reject the reservation, i.e., to generate a RESVERR message indicating a policy control failure. Instead, a new PATH message containing the new price is sent downstream. Now R1 has to decide whether it is willing to pay this higher charge and if yes, it sends another RESV message, which is treated similarly to the reservation from R2.

4 Assessment of the Charging Approach

The development of current Internet technology was largely driven by the aim to provide the best and simple technical solution for a given problem. However, for the Internet to evolve into *the* integrated services network of the future, the problem of appropriately charging users for communication services will be an important issue. Unfortunately, due to the history of the Internet and due to its historical funding structure, charging issues were never seriously considered when designing communication protocols. Therefore, in this section it is explained why some of the aforementioned expectations and requirements cannot easily be met by Internet- and particularly RSVP-technology.

Predictability of Charges. Because of the dynamics of RSVP and IP Multicast, prices can be predicted only to a limited degree, even for very short periods of time. Actually, when the reservation is supposed to be installed, the pricing situation might be completely different to what was announced to the receiver, as discussed in

Section 3.5. This uncertainty is analogous to the situation when using OPWA (One Pass with Advertising) [SB95]. Using OPWA, the AdSpec object advertises a QoS that the network offers to deliver, but this QoS might not be available anymore when the reservation shall actually be installed. Thus, we conclude that this behaviour is RSVP-inherent and is not due to our charging approach. To give an a priori specification of what a certain reservation over its whole duration will cost is not possible due to the soft-stated nature of our approach, which however in turn is 'inherited' by RSVP. It has to be accepted that prices change (in both directions) during a session.

However, such changes are always propagated to the user (in PATH messages) and have to be approved (by a RESV message). Thus, a receiver has control over its expenditures, but might potentially be frustrated by getting policy control failures due to short-term price changes. Despite the fact that prices are neither predictable at the start of a session nor constant during the whole session, they are however stable with respect to an RSVP session in equilibrium, i.e., when no one joins or leaves the multicast group and reservations and routes do not change. Furthermore, we assume that minor changes in reservations or group membership only lead to little pricing variations, i.e., prices change rather continuously.

Assuming this continuity, delivering the price information valid at the point in time when the PATH message is transferred to the users should give a fairly good estimate of the applicable price when a corresponding RESV message leads to a reservation. By introducing and delivering a certain maximum price in the DCPE we are also giving an upper bound on the charges – a worst-case price.

Stability of Service. In an integrated services network the precision of QoS predictions highly depends on the definition of service classes. In general, stability of service is rather an issue that is raised by the introduction of fine-grained charging than being a requirement for the charging scheme itself. It seems possible to add a refund mechanism to our charging approach, for example, by delaying the final accounting step until a communication service request is completely fulfilled, i.e., accounting is only done temporarily until charging state is orderly torn down.

Transparency and Accuracy of Charging. Transparency is the reason why we deliver the total charges all the way up to the sender. Thereby we enable a potential higher level protocol between sender and receiver – which will probably cooperate – to at least find out whether any provider is cheating. Some users might however desire more detailed information like, e.g., how one's reservations have been merged, what others pay, or the number of hops on the communication path that do not support RSVP respectively the requested service class. In that case we suggest to use the proposed RSVP diagnostics facilities [ZT97] or some extension of these procedures specialized on charging information. We perceive however that the availability of such information will possibly be restricted in a commercial environment. With respect to the accuracy or the level of detail of billing information that can be generated using our charging approach, it seems satisfying that each session can be billed separately.

Flexibility. We introduce flexibility by allowing both, sender and receiver payment for the communication service and even shared payments, thereby taking into account the diversity of communicating applications. Therefore, our charging approach can support the different application scenarios of value flow versus payment flow given in Section 2.1. Furthermore, little or no restrictions are imposed on the pricing policy of each network provider, thus, enabling a highly competitive environment. With regard to multicast communication, we observe that the collection of mechanisms proposed in this paper can be used to realize the different cost allocation strategies described in [HSE97]. Once again, we would like to emphasize that our approach separates mechanisms from policy/strategy. In particular, the frequency of price changes solely depends on each network provider's strategy.

Fraud Protection and Legal Security. The proposed charging mechanisms use the standard authentication methods provided by the RSVP framework to protect from misuse of a user's identity. A detailed discussion of this issue would be out of scope for this paper. Fraud protection is supported by transparency of charging information. Collaborating senders and receivers can compare the announced prices against each other and use the worst case price information for their charging limits. We mentioned the timing-related problems when establishing a reservation. It is an issue for further investigation how to handle the time gap between reservation initiation by a receiver and its establishment along the complete data path. Legal security, again, is rather a requirement that is introduced by precise charging in general, than being a requirement for charging mechanisms. In our approach, legal security is supported by transparency and accuracy of charging by giving at least some information that might be used as evidence in a litigation.

Technical Feasibility. Our charging approach has the same scalability characteristics as RSVP and does not increase its complexity. This is due to the fact that each UCPE and DCPE and its corresponding state can be mapped to the corresponding PATH and RESV state. While RSVP's scalability is currently under heavy discussion, the charging mechanisms at least do not add further complexity in the RSVP state management of routers. The amount of data exchanged for the set up of reservations is increased only moderately.

A similar, yet much simpler, approach to charge for RSVP flows is described in [FSVP98]. Its implementation is reported to add 0.75% protocol overhead and 2.3% execution overhead to RSVP processing. Although those mechanisms have significantly less complexity (achieved by covering a rather small set of charging scenarios), the given numbers are certainly an indication that detailed charging of RSVP flows is technically feasible.

Convenience. As already explained in Section 2.2 this requirement is mainly addressed by use of the edge pricing paradigm. Further, due to the abilities of senders and receivers to specify their maximum payment willingness, users may control their overall expenditures.

5 Summary and Future Work

In this paper, we described the basic layout of a charging approach for RSVP-based QoS reservations. We introduced a formal framework to model the flow of price information and payments and built charging mechanisms from this framework, that can be embedded into RSVP. It turns out that our approach supports most of the requirements to a charging scheme, whereas almost all restrictions and insufficiencies are inherited by the design of RSVP. Some of these restrictions might be overcome by augmenting certain RSVP messages or by using higher level protocols (similar to RTCP) to support charging cooperation between end systems. For example, a modified RESVCONF message could be requested by a receiver to gather detailed information about its reservation's status along the data path or at least until the first merging point, because of space limitations. In order to support the establishment of flows over the least expensive data paths, research work about QoS routing has to be carried out. This work must be extended by a new dimension: charge per QoS for a link.

Another important open research issue is the question how to flexibly represent prices and price variations for different requests within a single service class. The most flexible representation would be a price curve depending on the service class' traffic and QoS parameters. The representation of prices also influences the strategy how charges are split for merged reservations. However, it is not clear what level of complexity is introduced by such an approach. Furthermore, additional research is needed to understand the dynamics of pricing, payment methods and security issues.

References

[Bak97] Fred Baker. RSVP Cryptographic Authentication. Internet Draft, August 1997. Work in Progress.

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

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

[Her96] Shai Herzog. Local Policy Modules (LPM): Policy Control for RSVP. Internet Draft, November 1996. Work in Progress.

[Her97] Shai Herzog. RSVP Extensions for Policy Control. Internet Draft, March 1997. Work in Progress.

[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.

[Mor98] Sian Morgan. The Internet and the Local Telephone Network: Conflicts and Opportunities. *IEEE Communications Magazine*, 36(1):42–48, January 1998.

[NB98] Kathleen Nichols and Steven Blake. Differentiated Services Operational Model and Definitions. Internet Draft, February 1998. Work in Progress.

[SB95] Scott Shenker and Lee Breslau. Two Issues in Reservation Establishment. In *Proceedings of ACM SIGCOMM '95*, August 1995.

[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, Los Alamitos, CA, USA, June 1995. IEEE Computer Society Press.

[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.

[ZT97] Lixia Zhang and Andreas Terzis. RSVP Diagnostic Messages. Internet Draft, December 1997. Work in Progress.