A Concept for RSVP over DiffServ

R. Balmer, F. Baumgarter,T. Braun and M.Günter
Institute of Computer Science and Applied Mathematics, University of Berne, Switzerland
http://www.iam.unibe.ch/~rvs

Abstract— **Currently two approaches to provide Quality of Service in the Internet are being discussed: An early one is the Resource Reservation Setup Protocol (IETF RFC 2205) based on an end to end approach and on the other hand the recently ongoing activities in the IETF's Differentiated Service Working group focusing on methods for providing Quality if Service in backbones. This paper presents a concept for the integration of both Integrated and Differentiated Services, describes a prototype implementation, and presents evaluation results. Additionally the paper discusses business aspects arising from this service translation.**

I. Introduction

In the Internet of today there is an ongoing discussion about realising Quality of Services. One approach to achieve this was the development of the Resource Reservation Setup Protocol. This protocol is based on the idea of reserving resources for each TCP or UDP flows, causing every RSVP capable router to store information about this flow, to allocate resources and to instantiate traffic control components and queueing systems. Even when this works fine in small and medium sized networks, RSVP cannot scale in Internet backbones. On the other hand RSVP is really able to guarantee bandwidth and delay on a per flow basis, fitting the needs of modern real time applications.

The alternative concept for a Quality of Service supporting Internet are the so called Differentiated Services (DiffServ - IETF RFC 2475). The basic idea is the implementation of different traffic classes in the Internet. The differentiation among these classes is done by the Differentiated Service Code Point (DSCP) in the ToS byte of IP packets. According to the DSCP a packet will be put to queues with different priority or dropping algorithms (e.g. see IETF RFC 2597 and 2598) causing different packet forwarding. Every DiffServ capable host or network may apply – according to the established Service Level Agreement (SLA) – certain types of service to the packet leaving his domain. The SLA is a contract between two parties about the amount and type of traffic a party is allowed to send, respectively the Quality of Service the other party has to provide. It is obvious that the performance of Differentiated Services depends crucially on a good network provisioning within the backbone.

II. Basic Concepts of IntServ-DiffServ Mapping

To combine the advantages of Differentiated Services (good scalability in the backbone) and of RSVP (de facto standard, application support) a mapping from the RSVP reservation to an appropriate Differentiated Service class has to be performed. Two alternatives for interoperability between IntServ and DiffServ are mentioned in IETF RFC 2475.

The first option assumes to run IntServ and DiffServ independently of each other. Some flows such as real-time flows might get an IntServ reservation while others are supported by DiffServ mechanisms. This operation is simple but limits the use of RSVP to a small number of flows. In this mode, each node within the DiffServ network may also be a RSVP capable node.

The second approach assumes a model in which peripheral stub networks are RSVP and IntServ aware. These are interconnected by DiffServ networks that appear as a single network link to the RSVP nodes. Hosts attached to the peripheral IntServ networks signal to each other per-flow resource requests across the DiffServ networks. Standard RSVP processing is applied within the IntServ peripheral networks. RSVP signaling messages are carried transparently through the DiffServ networks. Devices at the boundaries between the IntServ networks and the DiffServ networks process the RSVP messages and provide admission control based on the availability of appropriate resources within the DiffServ network.

This model is based on the availability of services within the DiffServ network. Multiple Integrated Services microflows which exist in peripheral networks are aggregated into a behaviour aggregate at the boundary of the DiffServ network. When a RSVP request for an Integrated Service arrives at the boundary of a Differentiated Services network, RSVP style admission control is applied based on the amount of resources requested in the IntServ Flow Spec and the availability of DiffServ at the corresponding service level. If admission control succeeds, the originating host or the aggregating router marks packets of the signaled micro-flow according to the appropriate Differentiated Service level. The RSVP/IntServ over DiffServ is especially suitable for providing quantitative end-to-end services. The use of RSVP signaling provides admission control to the DiffServ network, based on resource availability and policy decisions.

There are a couple of central requirements for such an approach.

- A parallel operation of IntServ and DiffServ should be possible in the access network.
- The approach of mapping RSVP to a more scalable kind of resource reservation, should not be limited to Differentiated Services, because the method of resource reservation in the backbone may vary. In addition to the favored Differentiated Services a mapping (and aggregating) of different RSVP flows to ATM PVCs may be chosen as well as a mapping to different IP tunnels.
- The used architecture should not require a modification,

neither of RSVP capable applications nor of the end systems' RSVP daemons.
• The technology used for resource reservation in the backbone should conform to the standard Differentiated Services framework as described in RFC 2475.

III. RSVP Signaling and it's Extensions

This mapping task can be split into two parts. The first one is the RSVP signaling, which is of course used for the resource reservation in the access networks and also for triggering resource reservation in the ISPs. The second one is the technique of aggregating flows and reserving bandwidth inside the ISP's networks.

We propose a central instance in the ISP called Bandwidth Broker (BB) [Int99], [TWOZ99], which can be queried whether there is bandwidth available within the ISP's network and which supervises the ISP's resource management. How an ISP finally allocates resources is left to the ISP. We propose Differentiated Services for the resource management in the ISP's backbone, another choice might be the mapping and aggregation of RSVP reservations within ATM VCs. The ultimate goal of RSVP/DiffServ integration is to avoid any RSVP resource reservation between the ISP's border routers. The information about permissions and Service Level Agreements are located in the Bandwidth Brokers database. For a mapping the BB has to be queried, whether the reservation can be set up. So there is a need for interaction between the mapping component and the BB.

In subsection III-B we introduce a non-intrusive modification to RSVP signaling to meet the requirements of the BB approach. The method reacts directly on single reservation requests using the *resv* message.

A. RSVP signaling

First of we will give a short overview about standard RSVP signaling. RSVP is used to negotiate and set up a resource reservation for a specific flow. So, in every RSVP capable router information about the flows have to be stored, leading to the above mentioned scalability problems in backbone routers.

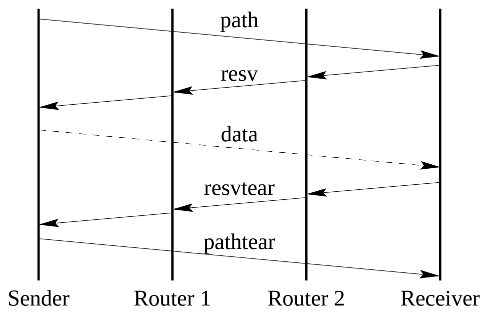

Fig. 1. RSVP Resource Reservation

Figure 1 shows the setup of a resource reservation. A RSVP session starts with a *path*-message being sent from the host, which wants to transmit data to the destination. This message has the following tasks:

• determination of the route the data will take later through the network[1].
• information about the destination, the traffic characteristics and perhaps the costs
• initialisation of information in each RSVP capable router on the path. Each router has to know it's neighbour routers.

If the receiver agrees the to reserve resources for the advertised flow, he sends back a *resv*-message, which is transported from hop by hop via RSVP capable routers towards the sender of the *path*-message. A RSVP router allocates resources and forwards the *resv*-message if he can meet the flows requirements, otherwise he replies an *resv-err*-message back to the sender of the *resv*-message.

If the receiver gets the *resv*-message, resources are reserved and the data can be transmitted. To terminate a reservation, an *resv-tear*-message is transmitted to remove the resource allocations and a *path-tear* message is sent to delete the path states in every router on the path.

B. Bandwidth allocation using the RSVP resv-message

Currently, DiffServ reservations are set up by the sender. Thus, the mapping from IntServ to DiffServ signaling has to take place at the border router on the RSVP initiator's side (sender of the *path* message). This can be done at two particular points in the protocol, namely at the arrival of the *path* message or at the arrival of the *resv* message. Since the *path* message is only preliminary and has other functions than establishing a reservation, the arrival of the *resv* message is the natural interception point. (see figure 2). The following steps describe the extended protocol:

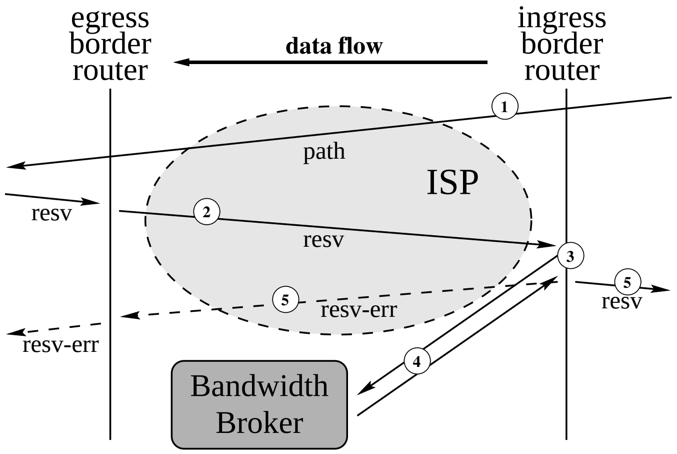

Fig. 2. Intercepting the *resv*-message

1. The ISPs ingress router receives the *path*-message and forwards it normally to the next router.
2. After the *path*-message has reached the receiver, a *resv*-message is generated and transported hop by hop to the sender of the *path*-message. It is assumed, that there are

[1] to achieve this the path message can not be sent hop by hop by changing the destination addresses to the next intermediate routers, but has to be transported directly to the destination. To force a processing in each forwarding router, RSVP uses the Router Alert Option in the IP header (IETF RFC 2113)

no RSVP capable routers in the ISP's backbone or the processing of RSVP signals is omitted by setting up tunnels between the border routers.
3. The *resv*-message reaches the ingress border router. This router is now responsible for reserving resources between the two border routers. So it queries the BB, whether the flow conforms to the SLA or not.
4. The BB decides upon the reservation, replies the result to the ingress border router and sets up appropriate resources within the ISP's network.
5. If the reservation is accepted, the *resv*-message is forwarded, else an *resv-err*-message is generated and replied to the sender of the *resv*-message.

This is the scheme used in our implementation.

IV. Implementation

For the prototype implementation commercial routers have been used. To provide QoS inside the ISP's network tunnels with a certain QoS shall be setup between the border routers. Because so far commercial router are missing RSVP-BB signaling each router was supported by a Linux router running a modified RSVP daemon to manage the signaling. Figure 3 shows the equipment and the topology of the test and demonstration network.

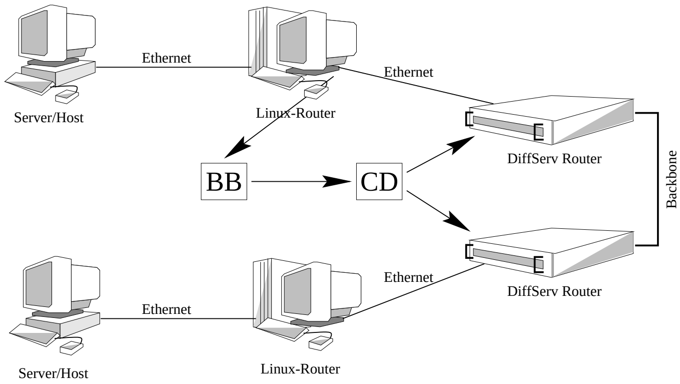

Fig. 3. Single ISP Scenario with two DiffServ and two Linux routers

The two DiffServ routers together with the Linux ingress routers realise the ISP's border routers. For QoS provisioning within the ISPs network the routers use VPN tunnels based on the concept of Differentiated Services (IETF RFC 2475). The two Linux machines outside the DiffServ routers have to keep track of RSVP signaling, the reservation of local resources and the interaction with the BB, which configures/monitors the DiffServ routers. The configuration daemons (CD) between the BB and DiffServ routers are used as some kind of adaptation layer. So the BB can use a "platform independent router configuration language" to configure the DiffServ routers. This shall allow the easy exchange of the router platform (see also [GBK99]).

Because of simplicity the implemented version of the RSVP DiffServ Gateway (RDG) directly connects to the ISP's BB. In reality the RDG may connect the BB of his local network, which then will negotiate with the ISP's BBs if necessary. The business aspects of the queried BB's location are discussed in section V.

The RSVP Daemon's Extension

As mentioned in section III-B the concept based on the use of *resv*-messages to trigger the ISP's resource reservations was chosen. From the Linux routers point of view the whole ISP network can be treated as a huge extension of it's local queueing system. The RSVP software we used (see [Ins], [Alm]) as a basis for our implementation was designed for a high portability to different router platforms, so there is a suitable interface between the queueing system (under Linux based on the programs/libs tc and ip) and the RSVP-daemon itself.

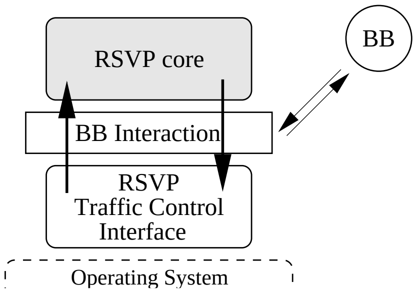

Fig. 4. Extension of the RSVP daemon

We used this interface to add the required functionality for the BB interaction. So every time local resources are reserved, released or modified also some routines are called, querying the Bandwidth Broker as shown on figure 4. By this a reservation is only successful, when the local traffic control system and the BB agreed. Another advantage is the full transparency of the extension, so no RSVP user outside the ISP will have to change anything or even consider a RSVP to DiffServ mapping occurred.

Interaction between RDG and BB

In [BCD+99] the RSVP objects are directly forwarded to the COPS server. The RDG presented here extracts the required information out of the RSVP objects and uses a couple of human readable commands to communicate with the BB. This simplifies debugging during development and performance evaluation. It also enables a simple interaction between different platforms. The actual used set of commands contains terms for user authentication, setup, deletion and modification of reservations. Another advantage of this simple protocol is the easy conversion from RDG requests to router configuration commands the Bandwidth Broker has to perform. Finally it should be mentioned that there is no high level functionality in the Linux boxes, because only the BB is responsible to decide how the request has to be handled. These open issues include:
Policy Control: Is the source or destination permitted to perform the request? The RDG may of course store some information about the SLA's of the according user to prevent BB interaction for each single reservation and do some

over provisioning in allocating resources to gain more local autonomy.

Charging and Accounting: How much has the user to pay for the reservation? How is the payment handled?

Reservation Method: In most cases the reserved flows are aggregated to large tunnels between the border routers, but it is also possible to map a reservation to an own tunnel.

Resizing of tunnels: The RDG might have different strategies in allocating resources at the ISP. So the RDG might not query the BB for every single reservation, but allocate more bandwidth than actually needed. So he can meet reservation requests without negotiating the BB (see section VI).

Bandwidth Reservation in the Backbone

The mechanism used to communicate between the RDG and the BB are completely independent from the concepts used for the final bandwidth reservation in the backbone. IP tunnels with a certain QoS are established between the two routers of the demo scenario. When the BB receives a reservation request, it reconfigures the router to transport this flow's data through a tunnel with the appropriate QoS.

V. Business Aspects

In parallel to the technical issues of mapping IntServ to DiffServ control messages, we must also consider the two kinds of services and their mapping from the business point of view. In this section we will discuss the fundamentally different business philosophies of IntServ and DiffServ, describe a unified scenario and how a particular IntServ payment mechanism can enhance the DiffServ architecture.

A. Cost and Price of the Services

Both DiffServ and IntServ reserve resources. While IntServ reserves them explicitly using reservation messages, DiffServ does so implicitly. Bandwidth brokers negotiate service level agreements (SLA) and use local mechanisms to grant the agreed per-hop-behaviour for in-profile traffic. Therefore, they have to reserve network resources. Of course, no provider will reserve resources without incentives. These come from the users demand for QoS-enabled Internet services. Voice and video applications need at least minimal guarantees from the network services. Since the Internet is still often congested, customers are willing to pay for resource reservation to support such applications [EV99]. Furthermore, the providers need value-added services since the prices for simple network connectivity services tend towards zero.

Although both IntServ and DiffServ generate value by offering resource reservation, they differ fundamentally when it comes to the question who pays for it. The Integrated Services follow a model inspired by large contents providers such as TV or radio broadcast stations. This model also applies to popular Web servers. The sender announces its service (RSVP PATH message) and the receiver 'subscribes' by setting up a reserved route to the sender (RSVP *resv*-message). This implies that the receiver has to pay for the reservation.

In DiffServ the situation is different. Before DiffServ packets can successfully travel through a provider network a SLA must be set up describing the traffic profile. Of course such a SLA will generally not be for free. Thus, DiffServ currently bases on a postal system model, where the sender pays. Provided that payment systems for IntServ and DiffServ will establish, the former will probably feature receiver payment, while the later will feature sender payment. An IntServ to DiffServ mapping must also address this fact.

Another problem is the granularity of the payment. IntServ will probably support micro-payments that must be handled electronically. DiffServ on the other hand has been designed to be scalable and implemented incrementally. The payment mechanisms will start with traditional off-line macro payments. With automation of SLA establishment (see e.g. [GBK99]) this may change in the future, but still the payments will be aggregates. When mapping IntServ to DiffServ not only traffic must be aggregated there, but also the charging and finally the payments. The following sections address the question how and where this should be done.

B. Location of Aggregation

IntServ is designed to accommodate the users' applications. DiffServ is designed to accommodate the backbone providers' needs. We can thus safely assume, that IntServ will be deployed only in the customer networks and maybe the access provider network, while DiffServ will be deployed in the large provider networks that form the Internet core. The aggregation is performed at the border of the IntServ and the DiffServ cloud. There are basically two options here. Either the aggregation is done on the customer (IntServ) side or at the provider (DiffServ) side of the border.

If the customer aggregates the traffic, it operates the RSVP-DiffServ gateway (RDG). The RDG notifies the local bandwidth broker (BB) of the customer. When a new reservation arrives that exceeds the profile described in the SLA, the BB has to renegotiate with the provider's BB. The customer (the sender) can also perform local admission control and check each RSVP reservation before it pays to set up the DiffServ reservation.

If the provider operates the RDG, this RDG signals the providers BB to set up the local resource reservation and to check the SLA with the involved neighbour providers. If necessary, the provider initiates the DiffServ reservation process. Therefore, the provider must probably pay on behalf of its customer (which is again the sender in the IntServ scenario). Of course, no provider would do that without a guarantee that the customer refunds these costs. Furthermore, the provider must manage the admission policies for the customer. It would be problematic if the provider sets up reservations upon an incoming RSVP reservation message, but later the customer rejects the reservation due to local policy considerations. While the aggregation of IntServ reservation over DiffServ can be a new provider service, the aforementioned problems of refunding mechanisms and policy control make the customer-controlled RDG option clearly more favourable.

C. Layered Payment Mechanisms

The DiffServ paradigm implies, that the sender pays for core network reservations. In many cases this is fine, namely when the sender has an economical interest to disseminate information with assured communication quality (e.g. commercials). However in other cases it is the receiver that demands QoS. To resolve the problem we must consider why the different payment paradigms emerged. It is because different service levels are mixed. Providers demand payment for pure connectivity. The higher a provider is in the hierarchical structure of the Internet, the more connectivity it provides to its customer. Therefore, a backbone provider will charge whoever sends packets through it.

This is in contrast to contents providers. The packet that they send contain information that represent a value for the receivers. Therefore, the receiver is willing to pay the transport costs. The resource reservation is something in between connectivity and contents. Sender *and* receiver payment paradigms exist for it, depending on whether the designers are guided by the connectivity- or the content provider paradigm. To resolve the conflicting payment paradigms we propose to decompose payment mechanisms into the following three layers.

- The lowest layer is connectivity payment. A customer generally has to pay to get connected to a network with Internet access. The payment for connectivity is mostly based on flat fees and seems to drop slowly towards zero.
- The next payment level is resource allocation. The price there is dependent of the amount of resources reserved, and maybe also of the concrete usage of resources.
- The top payment level is payment for contents. Generally the receiver has to pay for contents. This payment level is not necessarily directly related to network traffic anymore.

The higher levels generally deal with a higher price per bit. Therefore, the seller in a higher level is often willing to pay the lower level costs for the corresponding buyer. The higher level price will then include the lower level costs. Therefore, with a receiver oriented connectivity payment, a sender oriented reservation and a receiver oriented contents payment system, most of the useful scenarios can be implemented. Note, that in DiffServ, SLA brokering agents provide such a resource reservation payment mechanism, but for contents charging, other emerging technologies are necessary.

Using the contents level to share the resource reservation cost is a viable way, but introduces some overhead in the price calculation. A built-in support for cost sharing for the resource reservation layer is desirable.

D. A Unified Business Scenario

During the CATI project [SBGP99], the IntServ resource reservation mechanism has successfully been extended with a resource reservation payment mechanism, that allows cost sharing between sender and receiver. The main idea is that RSVP reservation messages contain also payment objects (digitally signed checks). Providers on the path extract those checks that are addressed explicitly to them. For more details on see [SFJ$^+$99]. To our knowledge, no similar mechanism exists for DiffServ. However, the IntServ-based CATI payment mechanism can be used together with the IntServ over DiffServ solution presented in section III.

Our solution thus allows the sender to share DiffServ reservation costs as depicted in Figure 5. As described earlier, the whole DiffServ cloud is interpreted as one hop for the IntServ messages. In the DiffServ cloud these messages are just forwarded unchanged. In the IntServ price sharing negotiation, the sender calculates the cost for the reservation in the DiffServ cloud by queries to bandwidth brokers, and announces the results to the receiver. Furthermore, also the cost-sharing is negotiated. The receiver will now add payment objects for each hop and send them periodically (1). In the scenario that we previously discussed, there will be just one big IntServ hop, namely the DiffServ cloud. At the sender site, the RDG intercepts the message. Now, not only admission control and aggregation is performed, but also the payment objects are extracted (2). The bandwidth broker of the sender now stores the payment objects (3). If the DiffServ payment mechanism supports the payment objects, the BB can pay aggregated DiffServ reservations with them (4). Otherwise, it must use a separate account to furnish the DiffServ payment mechanism and refill this account by periodically cashing the stored payment objects. By using our IntServ over DiffServ mechanisms and the IntServ reservation payment we can thus also extend the sender based DiffServ payment by a cost sharing option.

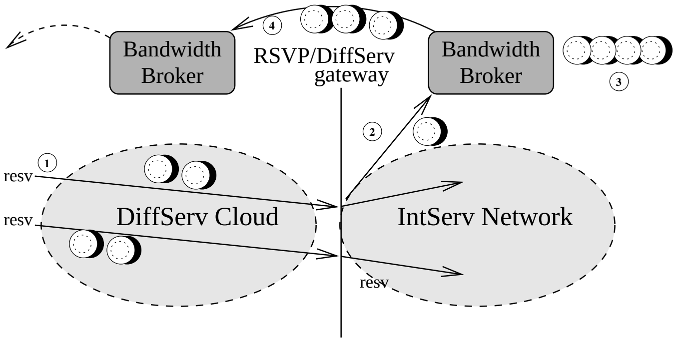

Fig. 5. Cost sharing using an IntServ payment mechanism.

VI. Results

In this section we will briefly present some results regarding the influence of network over provisioning and the performance of bandwidth reservation.

A. Signaling

A mentioned in section IV the RDG does not have to negotiate each resource request with the BB. It is a more favourable strategy to request more bandwidth than actually necessary to be able to answer requests locally without

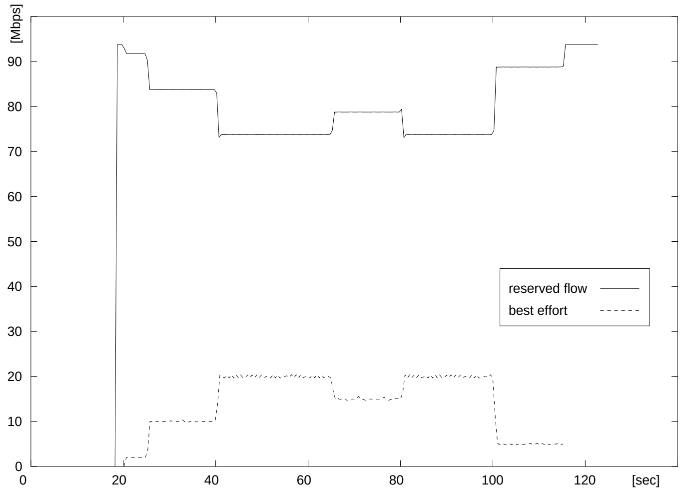

Fig. 6. The effect of reservation

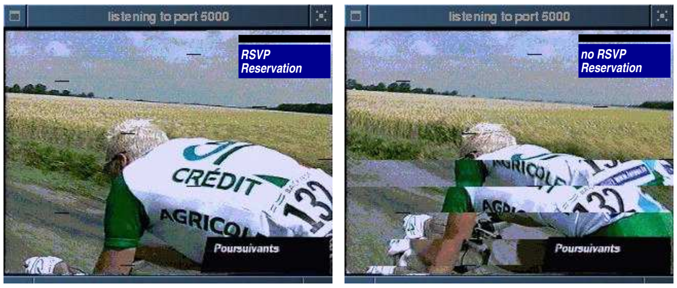

Fig. 7. mjpeg video transmission with and without bandwidth reservation

querying the BB. Simulations in [DGBS00] showed that an over provisioning of 10 percent already reduces the amount of DiffServ negotiations to less than the half.

B. Bandwidth Measurements

The graph on figure 6 shows the achieved bandwidth of an UDP flow, transmitted over a RDG through an Differentiated Service domain as shown on figure 3. The resource reservation was triggered by RSVP requests with several modifications of the reservation. The flow got the reserved bandwidth despite a parallel UDP flow causing heavy congestion. As expected the flow got the requested resources. The two pictures on Figure 7 show the influence of bandwidth reservation on a mjpeg video transmission. The left picture was transmitted with, the right one without bandwidth reservation. The picture quality decrease, resulting from missing UDP packets is obvious.

VII. Conclusion and Outlook

Integrated Services allow per micro-flow resource allocation, but do not scale to the core Internet. Resource reservation in the core is more likely to be deployed using the Differentiated Services architecture. However, the requirements of user applications are better met by IntServ. To address this conflict we present a prototype implementation capable of mapping RSVP reservations to DiffServ reservations. An extended RSVP daemon (the RSVP-DiffServ gateway - RDG) monitors the RSVP *resv*-messages and contacts the appropriate bandwidth broker, which is a kind of bandwidth manager of a DiffServ domain. The bandwidth broker can thus map IntServ reservations to an existing DiffServ reservation, trigger new DiffServ reservations or reject the IntServ reservation. Since reservations are not for free in general, we also considered the business aspects of mapping IntServ to DiffServ reservations. In particular the conflict of IntServ's "receiver pays" paradigm with DiffServ's "sender pays" paradigm is of interest. We identify this conflict as an instance of the more fundamental problem of dividing the price into a transport component and content component. However, in the special case that the RSVP *resv* signaling already contains payment objects (provided by the receiver), we describe how the bandwidth broker can collect and use them to pay for necessary DiffServ reservations. Furthermore, we argue that the RDG should be located at the edge of a host network so that it should contact the local bandwidth broker.

VIII. Acknowledgement

The work in this paper is part of the work in the project Charging and Accounting Technologies for the Internet (CATI) [SBGP99] funded by the Swiss National Science Foundation (SNF) (Project no. 5003-054559/1 and 5003-054560/1). The implementation platform has been funded by the SNF R. Equip project no. 2160-053299.98/1. The foundation 'Stiftung zur Förderung der wissenschaftlichen Forschung an der Universität Bern' funded also parts of the implementation platform and the publication of this paper.

IX. References

References

[Alm] W. Almesberger. Tc compatible linux version of the isi rsvp implementation, version 4.2a4. URL: ftp://lrcftp.epfl.ch/pub/people/almesber/rsvp.

[BCD+99] J. Boyle, R. Cohen, D. Durham, Sh. Herzog, R. Rajan, and A. Sastry. Cops usage for rsvp. Internet Draft `draft-ietf-rap-cops-rsvp-05.txt`, June 1999.

[DGBS00] Gabriel Dermler, Manuel Günter, Torsten Braun, and Burkhard Stiller. Towards a scalable system for per-flow charging in the internet. In *Proc. of the Applied Telecommunication Symposium (ATS 2000)*, 2000.

[EV99] R. Edell and P. Varaiya. Providing Internet access: What we learn from INDEX. *IEEE Network*, 13(5):18–25, September/October 1999.

[GBK99] M. Günter, T. Braun, and I. Khalil. An architecture for managing QoS-enabled VPNs over the Internet. In *IEEE Conference on Local Computer Networks*, 1999.

[Ins] Information Science Institute. Implementation of the resource reservation protocol. URL: ftp://ftp.isi.edu/rsvp-/release.

[Int99] Internet2 QoS Working Group. Qbone. http://www.internet2.edu/qos/qbone/, 1999.

[SBGP99] B. Stiller, T. Braun, M. Günter, and B. Plattner. Charging and accounting technology for the Internet. In *Multimedia Applications, Services, and Techniques*, May 1999.

[SFJ+99] B. Stiller, G. Fankhauser, G. Joller, P. Reichl, and N. Weiler. Open charging and qos interfaces for ip telephony. *The Internet Summit (INET '99)*, June 1999.

[TWOZ99] Andreas Terzis, Lan Wang, Jun Ogawa, and Lixia Zhang. A two-tier resource management model for the Internet. In *IEEE Global Internet'99*, December 1999.