Implementation of an Internet Video Conferencing Application over ATM

Torsten Braun, Heinrich J. Stüttgen
IBM European Networking Center, Vangerowstr. 18, D-69115 Heidelberg, Germany
Phone: +49 6221 59-[4352|4337], Fax: +49 6221 59-3300
Email: [braun|hjs]@heidelbg.ibm.com

Abstract. The paper describes a complete implementation of an Internet video conferencing application taking advantage of the quality-of-service (QoS) capabilities provided by ATM. RSVP has been used to exchange the QoS parameters among the communicating video conferencing applications. The RSVP QoS parameters are mapped to ATM parameters and dedicated VCs with the corresponding ATM paramters are established by the underlying classical IP over ATM implementation. These VCs are individually used by the application data flow and provide the desired QoS from the ATM network to the application. The implementation is based on existing components such as an Internet conferencing tool, a RSVP implementation, and a classical IP over ATM implementation. The paper presents the required modifications of all these components to support QoS communication and the adaptations for their integration.

1 Introduction

Videoconferencing and telephony are popular applications being used in the Internet. Many of these applications are adaptive. This means that they are able to adapt to varying network conditions such as congestion situations. In the case of congestions they are usually able to slow down the transmission rate by changing compression or encoding parameters. This avoids further congestions and avoids packet losses [13]. However, slowing down the transmission rate usually also means that the quality of the transmitted audio or video decreases. Often, the audio/video applications need a minimal bandwidth to provide a reasonable quality of the communication.

Adaptation mechanisms alone are, therefore, not sufficient to provide reasonable QoS to the users, in particular in frequently overloaded networks. Guaranteeing a minimal QoS requires ressource reservation techniques. In particular network ressources need to be reserved in order to provide QoS guarantees to applications running over the Internet.

To support QoS, the Resource ReserVation Protocol (RSVP) has been developed within the Internet world [5]. RSVP provides mechanisms to support QoS negotiations among communicationg applications and ressource reservations in end systems, routers, and networks.

One of the most important resource is network bandwidth. However, only a few network technologies such as ATM are today able to perform bandwidth reservation. ATM is becoming widely used either as backbone technology or as local network technology. Using ATM as network technology for the integrated Internet services, for which RSVP has been designed, requires to map RSVP and ATM to each other. This means mapping the QoS parameters of RSVP and ATM as well as mapping the RSVP and the ATM signalling protocol to each other.

This paper describes the implementation of a video conferencing application over RSVP and classical IP over ATM. All parts have been implemented on a RS 6000 workstation running AIX version 4.2.

The application has been modified in order to take advantage from the resource reservation capabilities of the underlying communication system. In particular, the user interface has been modified to allow the user to specifiy the desired video quality. The user QoS requirements are mapped to RSVP parameters which are exchanged among RSVP systems (end systems and routers). Dependent on the RSVP parameters the RSVP systems have to reserve the ATM network resources required by the application.

Section 2 describes the required modifications of a classical IP over ATM implementation in order to support QoS for IP flows. Section 3 presents the adaptation of the RSVP implementation to support resource reservation in classical IP over ATM subnetworks. Section 4 discusses the modifications of the video conferencing application in order to take advantage the RSVP/ATM based communication system. An evaluation of the complete system design and its implementation concludes the paper.

2 Implementation of IP over ATM with quality-of-service-support

ATM connection establishment following the classical IP over ATM concept [10] means that a single, shared best-effort virtual connection (VC) is established for several IP flows between any pair of ATM attached IP systems. The VC is established with quality-of-service (QoS) parameters supporting best-effort service according to [12] only. An IP flow is a sequence of packets exchanged among applications. It is identified by the so-called flow identifier. In IPv4, this is the combination of the source/destination IP addresses and the source/destination TCP/UDP port numbers. In IPv6 the flow identifier may consist of the flow label and the source IP address.

In a classical IP over ATM implementation, several applications have to share best-effort ATM VCs if they have the same next hop, i.e. the next router or the same destination if this is located in the same logical IP subnet. However, real-time applications with high quality-of-service requirements require dedicated VCs with individual QoS parameters for a single flow.

This section describes an implementation architecture to provide ATM QoS support for IP flows [7]. The architecture is based on a classical IP over ATM implementation, which has been extended to support QoS on application level. System calls have been implemented, which allow to specify traffic descriptor parameters to be used in SETUP messages for VC establishment. If a dedicated VC exists for a flow, the packets of the flow are sent over this VC, otherwise a best-effort VC is used, which must be shared with other flows.

The classical IP over ATM implementation extensions provide an interface to receive QoS information from the applications. This QoS information is used to establish individual ATM-VCs to be used for particular IP flows. The QoS information and the flow identification are stored within additional fields of ATM ARP table elements. This table is processed by the classical IP over ATM implementation for an IP packet to be transmitted. In a classical IP over ATM implementation, the ATM ARP table maps the destination IP addresses to the VC to be taken for transmitting an IP packet. To support QoS on IP flow level, flow identifiers must be mapped to VCs, which are associated with ATM QoS information. These so-called filter entries consisting of QoS information and the flow identifier can be written, read and deleted by using well-defined I/O control commands:

- The write command delivers a filter entry to the classical IP over ATM implementation. The ATM VC with the corresponding QoS parameters is established for the next packet of the specified IP flow to be processed. After the QoS-based VC has been established, it is exclusively used for the requesting application. The best-effort VC is kept up for other IP flows after the QoS-based VCs are established.
- The read command allows to check whether an ATM VC has already been established for a certain IP flow.
- The delete command deletes a filter entry from the ATM ARP table. Deleting the filter entry for a certain IP flow in the ATM ARP table results in the termination of the corresponding VC. Packets of the deleted flow will again be sent over the best-effort VC.

When a data packet for a certain flow has to be sent, the extended classical IP over ATM implementation looks if there is a filter entry for the flow in the ATM ARP table. If a filter entry is found and a VC does not exist, it tries to establish the VC by sending an ATM SETUP message. The QoS parameters of the ATM ARP entries, which have been configured by a write command, are used for ATM connection setup. The connection parameters, which are included in the QoS information structure, are copied into the corresponding fields of the SETUP message.

In the mean time the packets of the flow for which the connection is being established are sent over the best-effort VC to the destination. As soon as the dedicated VC is established, packets belonging to the flow are sent over the dedicated VC.

Handling of incoming calls had slightly to be modified, i.e. the receive part of the classical IP over ATM implementation had to be adapted. In a normal classical IP over ATM implementation an Inverse ARP message is sent back to the calling system in order to find out its IP address when a connection establishment request is received. Since the QoS-based VCs are unidirectional (as RSVP reservations are unidirectional), this Inverse ARP message will not be sent for QoS-based VCs. QoS-based VCs are identified by having the broadband Bearer Capability class set to C (for best-effort VCs the used class is X), and the backward sustainable cell rate set to 0 (unidirectional VC).

3 RSVP Implementation over ATM

The extensions to the classical IP over ATM implementation allow user-level programs to establish dedicated VCs for IP flows with specific QoS. This allows a resource reservation protocol to benefit from QoS guarantees across an IP over ATM subnetwork. In this section we describe how an RSVP implementation uses the classical IP over ATM extensions.

The ISI RSVP daemon [14] runs in user space and is designed to communicate with a traffic control kernel to perform the resource reservations. In order to make use of the features provided by the classical IP over ATM extensions, the RSVP daemon implementation had to be modified. The RSVP daemon, which has been designed for routers and end systems, has three major interfaces:

- Applications communicate with the RSVP daemon via the RSVP Application Programming Interface (RAPI) and an UNIX pipe [6]. The RAPI library routines linked to the application allow an application to create a RAPI session, to make reservations, or to delete them. E.g., a receiver application uses rapi_register to initialize a reservation and rapi_reserve calls to establish or modify reservations for the session. The rapi_release call deletes the reservations. The RSVP daemon generates, sends, receives, and processes RSVP messages dependent on the registered reservations.
- A general framework known as RSRR (Routing Support for Resource Reservation) is used as interface to routing protocols. It includes an interface to the multicast routing code and an unicast routing interface to the kernel routing table.
- The RSVP daemon communicates with a traffic control kernel through a traffic control (TC) adaptation module which is a part of the RSVP daemon. The RSVP daemon can be used to establish reservation states in hosts and routers. The daemon may pass these states to the traffic control kernel via the kernel-specific TC adaptation module. The packet classifier, packet scheduler and the admission control module can be considered as parts of a traffic control kernel.

The original RSVP daemon contained a TC adaptation module for the MIT ISPS kernel. In order to perform the additionally defined ioctl calls of the kernel extensions and to reserve ATM resources, a new TC adaptation module for ATM has been implemented. The RSVP daemon cooperates with the ATM TC adaptation module through several function calls. Their function interfaces have been nearly kept unchanged compared to the original release. The ATM TC adaptation module is the interface between the RSVP daemon and the classical IP over ATM extensions (Figure 2). The video conferencing application calls the RSVP daemon by RAPI calls. Then, the RSVP daemon performs function calls to the TC adaptation module. The TC adaptation module then establishes dedicated QoS-based ATM VCs through writing filter entries to the classical IP over ATM implementation extensions.

The TC adaptation module provides several interface functions. The RAPI call rapi_reserve at the receiver (host B) triggers the sending of a reservation message to the sender (host A). After receiving a reservation message the RSVP daemon at host A calls the traffic control function TC_AddFlowspec using the flow specification and traffic specification parameters. The function translates the Internet integrated services QoS parameters into ATM connection parameters. QoS mapping between IETF service classes and ATM connection parameters is described in [4]. The current implementation realizes a committed rate service [2].

The RSVP daemon then calls the TC_SetFilter function in order to associate the entry created by TC_AddFlowspec to a specific IP flow, and to update the filter entry to the ATM ARP table. The TC_SetFilter function performs a read call to the classical IP over ATM implementation in order to check whether a private VC already exists for this flow. If no filter entry is found, a new filter entry is written to establish the QoS-based VC. When the VC is fully established, packets belonging to this specific flow are sent over the dedicated QoS VC.

If the application at host B now changes its reservation (e.g., asking for a higher bandwidth by rapi_reserve), a new RSVP reservation message is sent to host A. At host A, the TC_ModFlowspec function is called by the RSVP daemon. This function first checks the existence of the specified filter entry within the ATM ARP table and deletes it. This triggers tearing down the private VC. Then it translates the new QoS parameters into ATM connection parameters and writes the new filter entry into the ATM ARP table. Then, the classical IP over ATM implementation establishes a new VC with the modified ATM parameters.

If the application at host B wants to delete a reservation, a RSVP reservation tear message is sent to host A. This implies that the RSVP daemon calls the TC_DelFlowspec function, which deletes the filter entry from the ATM ARP table. This also implies that the corresponding VC is torn down and further data of this flow are sent over the best-effort VC.

4 Adaptation of the Video Conferencing Tool to RSVP

The video conferencing tool (vic) [11] is one of numerous video conferencing tools developed for the use over the IETF multicast backbone (Mbone). We adapted vic for its use in an RSVP environment [8]. Currently, only unicast communication is supported.

The receiving user has the option to switch the reservation mode on or off by pushing the buttons “Reserve” and “RSVP STOP”. Pushing these buttons results in corresponding RAPI calls from the application to the RSVP daemon. Pushing “Reserve” results in a rapi_register call, pushing “RSVP STOP” results in a rapi_release call. When a reservation is set up at the sender, the actual transmission rate is advertised in the path message and displayed at the receiver (rsvp_info window in Figure 3, 1.8 Mbps).

The vic menu has been modified in order to allow the receiver to specify the desired bandwidth. Another slider was added to two exisiting sliders that allow the sender to specifiy the bit rate and the frame rate (Figure 3). The value of the reservation slider determines the QoS parameters in the RSVP reservation messages. Moving the reservation slider triggers a rapi_reserve call to indicate that the user desires new reservation parameters for the application data flow.

The reserved bandwidth is exchanged among the RSVP end systems by RSVP signalling messages. At the sender side, the bandwidth reserved by the receiver or the next router is displayed (rsvp_info window in Figure 3, 812 kbps).

5 Evaluation

The paper presented an Internet video conferencing implementation over ATM supporting QoS by ATM resource reservation. QoS mapping between the application and the ATM network is performed by using RSVP. Three components had to be modified in order to integrate them. All components use well-defined interfaces, which allows to substitute them if new components become available.

Only minimal modifications allowing QoS specifications have been performed within the video application. No modifications of socket calls to send and receive application data have been required. Mainly, some functions for QoS control had to be added to the application. Only a few existing modules needed to be modified.

The application allows receivers to specify the desired QoS. User QoS parameters are mapped to ATM QoS parameters and individual ATM VCs are established with the individual QoS parameters for the application data flow. QoS mapping is performed by the RSVP daemon running in user-space. This allows quick and easy adaptation of the mapping of integrated services to ATM traffic parameters when new Internet service classes will be specified or new ATM services such as ABR become available.

Multicast is currently not supported. The main reason is the lack of multicast functionality in the classical IP over ATM implementation. IP multicast over ATM may be added according to the MARS approach [1]. Multicast functionality can easily be integrated with the QoS extensions. The main problems and possible solutions in the case of multicast support are described in [3].

A QoS extension of the classical IP over ATM implementation has the main advantage over a native ATM socket based implementation that the application has only slightly to be extended and that heterogeneous network scenarios with mixed network technologies such as Ethernet, FDDI and ATM are supported. However, in a network where routers are interconnected via ATM, no direct ATM VCs can be established without using other protocols such as NHRP [9]. A native ATM socket based approach allows direct ATM connections between senders and receivers but limits the use to ATM networks. Furthermore, the required modifications of the application are higher than compared to the implementation presented in this paper.

ATM VC establishment and termination is completely transparent to the user. ATM connection management (establishing and terminating ATM connections) is handled by the classical IP over ATM implementation transparently to the applications running in user space. In a native ATM socket based environment, ATM connection management must be done by the application. Therefore, the user does not have to care about ATM connection management procedures and does not have to understand the meaning of the ATM traffic parameters.

6 Acknowledgements

The authors want to thank Stefano Giorcelli and Stefan Bretz for their contributions to the implementation of the architecture described in this paper.

7 References

[1] G. Armitage: Support for Multicast over UNI 3.0/3.1 based ATM Networks, RFC 2022, February 1996

[2] F. Baker, R. Guerin, D. Kandlur: Specification of Committed Rate Quality-of-Service, Internet Draft, November 1996

[3] S. Berson: IP Integrated Services Support in ATM, Internet Draft, September 1996

[4] M. Borden: Interoperation of Controlled-Load and Guaranteed Service with ATM, Internet Draft, September 1996

[5] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin: Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification, Internet Draft, October 1996

[6] R. Braden, D. Hoffman: RSVP Application Programming Interface (RAPI) for SunOS/BSD, Internet Draft, August 1995

[7] T. Braun, S. Giorcelli: Quality-of-Service Support for IP flows over ATM , "Kommunikation in Verteilten Systemen", Informatik aktuell, Springer-Verlag, 1997

[8] S. Bretz: Integration of a RSVP Resource Reservation Module into a MBone Multimedia Application, Diploma Thesis, BA Mannheim, September 1996 (in German)

[9] D. Katz, D. Piscitello, J. Luciani: NBMA Next Hop Resolution Protocol (NHRP), Internet Draft, September 1996

[10] M. Laubach: Classical IP and ARP over ATM, RFC 1577, January 1994

[11] S. McCanne, V. Jacobson: vic: A Flexible Framework Framework for Packet Video, ACM Multimedia '95

[12] M. Perez, F. Liaw, A. Mankin, E. Hoffman, D. Grossman, A. Malis: ATM Signaling Support for IP over ATM, RFC 1755, February 1995

[13] T. Turletti, C. Huitema: Videoconferencing on the Internet, IEEE/ACM Transactions on Networking, Vol. 4, No. 3, Juni 1996, S. 340-351

[14] URL: ftp://ftp.isi.edu/rsvp/release

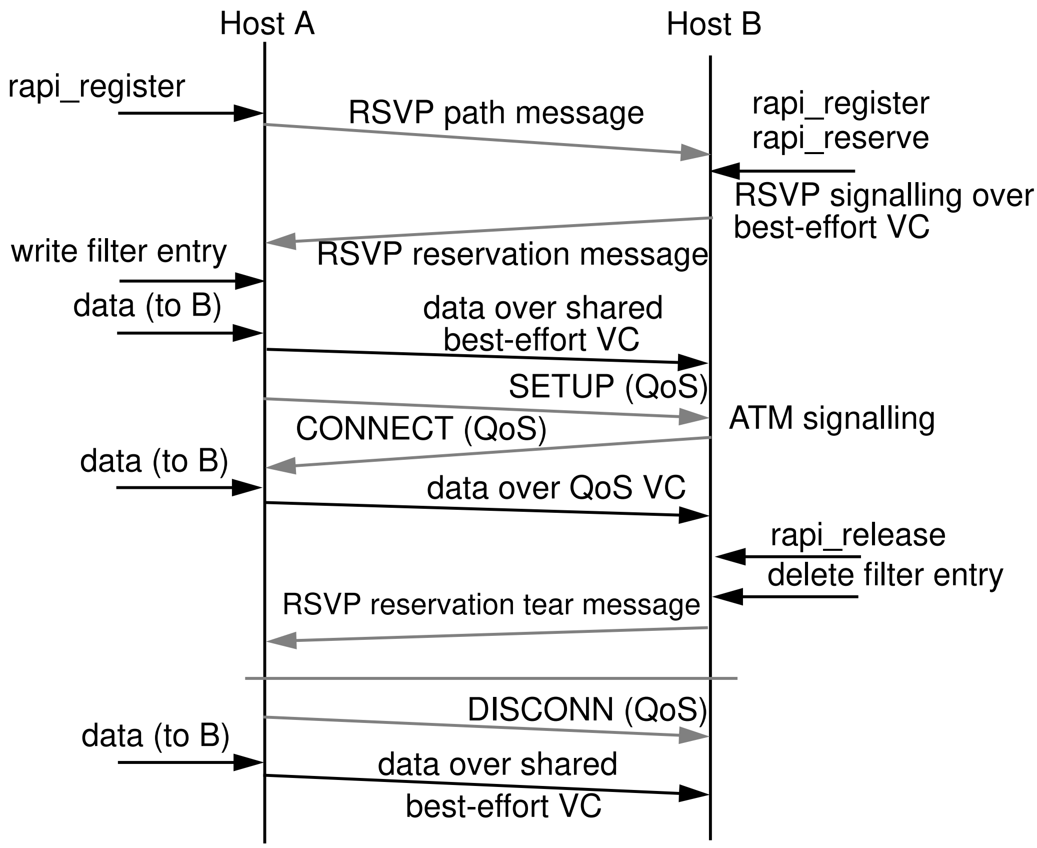

Figure 1: RSVP and ATM signalling

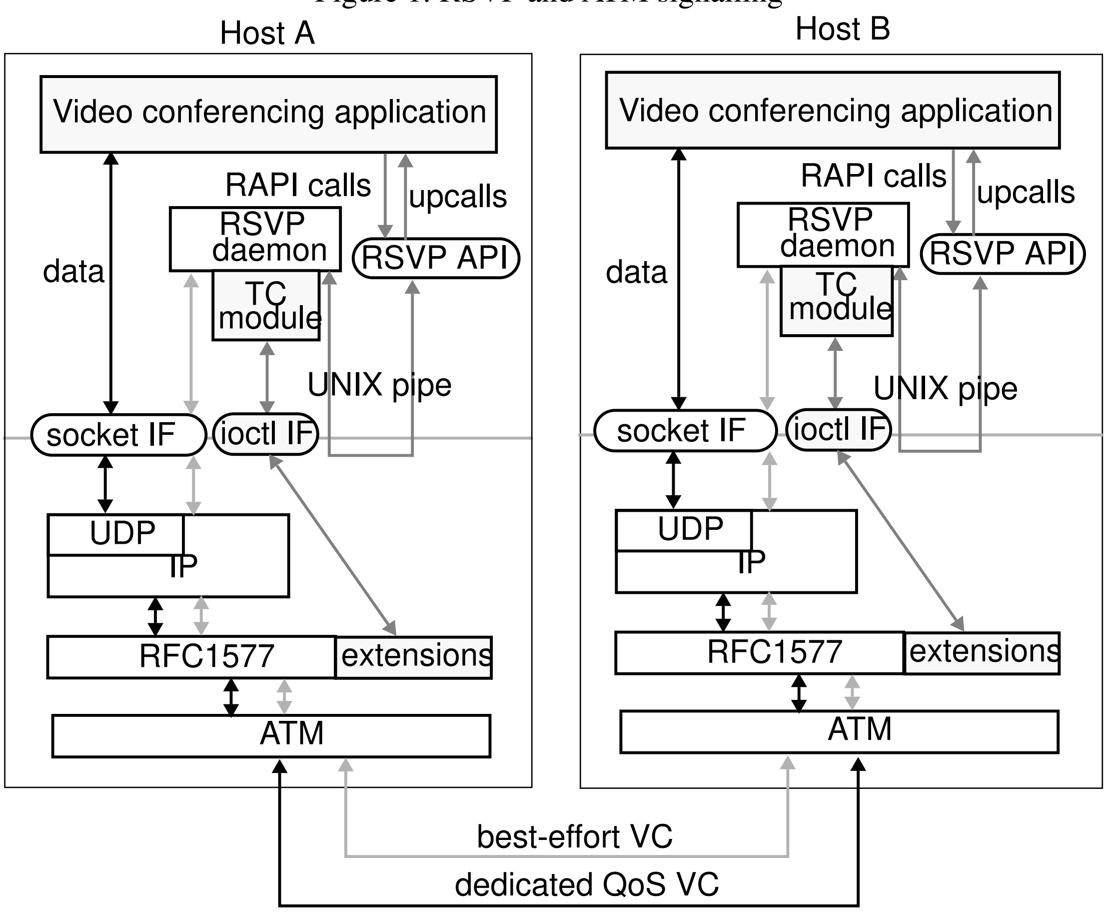

Figure 2: Overall Implementation Architecture

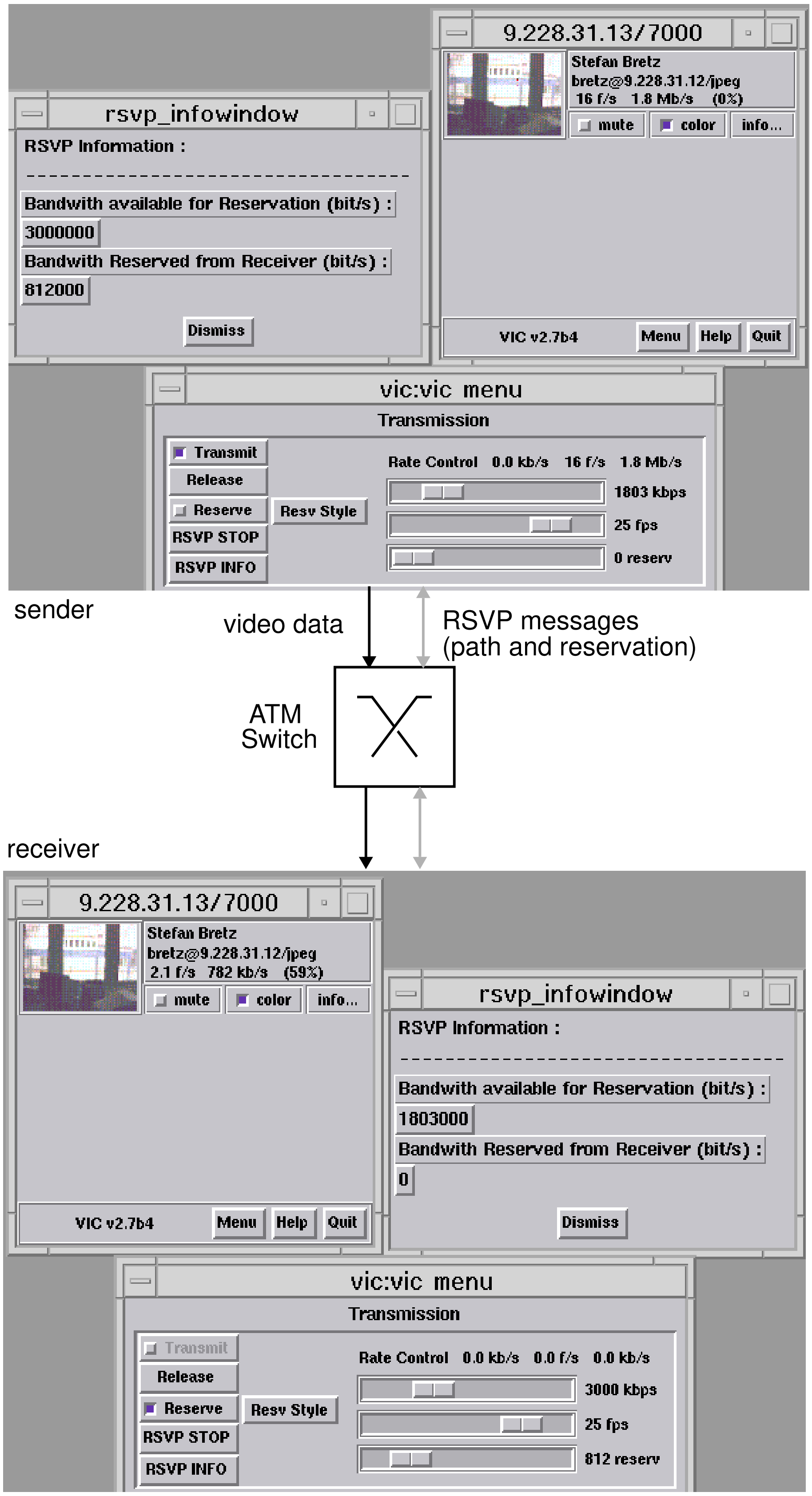

Figure 3: Video Conferencing Scenario