GCP

A Group Communication Protocol for Supporting Closed Groups in the Internet

Mario Zuehlke and Hartmut Koenig
Brandenburg University of Technology at Cottbus
Department of Computer Science

email: {mz,koenig}@informatik.tu-cottbus.de

Abstract Video conferences have become a popular application in modern high performance networks. Advanced technologies allow their installation on desktop computers. Currently videoconferences are mainly deployed for large open group conferences in the Internet to transmit lectures, conferences, talks, and discussions. Multiparty video conference tools for closed group meetings are less broadly used. On the other hand, most meetings in every-day life are meetings of closed groups with a more or less confidential character. Unlike open group conferences closed group meetings require a sender related transmission paradigm, i.e. the sender must know the receivers of the data streams. This is usually solved by using a centralized group server for the group and the QoS management. The shortages of this approach are server failures and performance problems. The decentralized approach is more flexible, but it requires more sophisticated mechanisms to preserve the consistency of the management data at each participant. In the paper we present the signalization protocol GCP to support decentralized closed group video conferences. It provides an atomic, ordered, and reliable data delivery service and supports a dynamic join and leave of participants. We present two versions of the protocol: one for ATM networks using overlaid multicast connections and one for IP-based networks.

Keywords: Multiparty video conferences, closed groups, group communication, distributed group management, ATM, IP

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35584-9_19](https://doi.org/10.1007/978-0-387-35584-9_19)

1. MOTIVATION

Video conferences have become a popular application in modern high performance networks. Due to improved network, processor, and compression technologies, and reduced equipment and network cost as well video conferences do not any more depend on specialized studio equipment. Video conference systems are available on desktop computers meanwhile. Despite of years of development, however, video conference services are not that broadly used like services such as e-mail or WWW.

Currently videoconferences are mainly deployed for large video conferences for transmitting lectures, talks and discussions in the Internet [1]. Such video conferences are open group conferences. The sending paradigm is receiver related multicast. Everyone who subscribes to the multicast address can join the conference. The sender does not know who the actual participants of the conference are.

Multiparty video conferences for closed groups, e.g. project meetings, teleseminars, medical consultations, are less deployed. Most meetings in every-day life, however, are closed group meetings with a more or less confidential character. Under a closed group we understand here a tightly coupled group with a sender related multicast paradigm, i.e. the sender always knows the receivers of the data streams. The participation in the meeting is by invitation. Depending on the agreed confidence of the meeting authentication as well as encryption of the data streams are required. Note that not all closed group video conferences, e.g. teleseminars, require this just as not every e-mail must be encrypted. Videoconference systems for closed group meetings are mostly based either on circuit-switching networks like ISDN using the H.320 standard or on H.323 based systems in packet-switching networks. Examples are CU-SeeMe and Netmeeting.

In wide area networks it is still difficult to ensure a good Quality of Service (QoS) at the participants using these standards. Many video conference applications over the Internet use instead the Multicast Backbone (MBone) running the MBone video conference tools or the USMInT service. These applications support open group conferences. It is possible to set up closed group meetings by encrypting the media streams, but the key management and the speed of encryption are insufficient. Even more advanced encryption functions do not provide a sufficient confidentiality. It remains the possibility for an attacker to secretly attach to a conference if the multicast address is known to record the (encrypted) conference, to disturb or attack it. On the other hand, it can be expected that a convenient access to closed group video conference services will lead to a rapid growth of their use. This will exceed the bandwidth capacity of the MBone routers. For

setting up closed group video conferences, it is therefore more appropriate to support the sender related multicast paradigm.

Closed group video conferences require a group management to supervise the composition of the group, to control the access to shared resources such as the audio channel and the white board (floor control), and to tune QoS parameters. For closed groups, this is usually solved by a group server which centrally manages the group and takes all decisions. The main shortage of the centralized approach are server failures that terminate the conference. The server may also become a performance bottleneck due to many server accesses. Decentralized solutions are more flexible in this case. They avoid the performance problem and do not terminate the conference in case of a participants' failure. The distributed approach, however, requires a more sophisticated design. So far only a few distributively managed video conference systems are reported [1], [2].

In this paper we present the signalization protocol GCP which has been designed to support a distributed group management in closed video conference systems. It provides an ordered, reliable, and atomic data delivery service. We present two versions of the protocol: GCP_{ATM} developed for ATM based video conference systems and GCP_{IP} designed for IP-based network environments. The paper is organized as follows. In Section 2 we discuss the requirements to the group management of distributed video conference systems and give a short overview of related work. In Section 3 we first show how an ordered, reliable, and atomic data delivery service can be provided in ATM networks using overlaid multicast connections. Section 4 presents a solution for this service for IP networks using point-to-point connections. The final remarks give an outlook on next research steps.

2. REQUIREMENTS TO THE GROUP MANAGEMENT

The group management supervises the composition of the conference. It stores all information about the participants and handles the requests to join and leave the conference. The group management is closely related to the QoS management of the exchanged data streams. It is also connected with the applied floor control strategy.

The group management depends on the nature of the conference. A strongly coupled conference, e.g. a meeting, has other requirements to the group management than a loosely coupled one like a chat. A conference that requires access permission (invitation, authentication) needs another group management than a large public conference.

Another essential factor for the design of the group management is the kind of its organization. Most existing video conference systems use a centralized group server. The server controls the whole conference. Usually, the initiator of the conference provides this function. When he/she leaves the conference mostly has to be terminated. Centralized solutions are easy to implement because all management functions are concentrated in one system. This is especially advantageous if sophisticated control algorithms are applied, e.g. for scaling video streams to the actual load of the end systems. The shortages of the approach are as discussed above server failure and performance bottlenecks. The alternative decentralized approach avoids these problems. It is more flexible, especially in larger networks like the Internet, but it is more complicated to design. One of the main issues of the distributed approach is to preserve the consistency of the group and QoS data at each participants' host to ensure a unique view on the conference state for all decisions. It further requires mechanisms for a distributed calculation scheme for scaling the QoS parameters and for assigning the floor.

In order to ensure the consistency of the group and QoS data at the participants a distributed conference management has to support the following features:

- **Ordered delivery:** The messages are delivered in the order they are sent. This ensures that events like joining or leaving of a conference, or assigning the floor are indicated to all participants in the order they occur. Depending on the application different levels of ordering (causal order or total order) can be applied.
- **Reliability:** In contrast to the transmission of continuous data, where some frames may be lost, distorted, or discarded, signalization messages have to be delivered reliably.
- **Atomicity:** It must be ensured that the messages are either delivered to all participants or to none of them. This property ensures that all participants are equally updated.

Furthermore, it is desirable to have a dynamic group management which allows the participants to join and leave the conference at any time, and to exclude participants due to system failure. The protocol should be simple to limit the signalization overhead.

The mostly used transport protocol for multimedia data transmission in the Internet, the RTP/RTCP [14], does not fulfill these requirements (and it also was not its intention). The associated *Real time Transport Control Protocol* RTCP provides a feedback by periodically sending *Receiver Report* messages between the communication partners which inform about group membership and the quality of the data transmission. Thus RTCP only provides approximate information on the system and group state [1]. A

group member never exactly knows who is in the conference at a given moment. Moreover the leaving of the conference by a BYE message can cause error situations. When a participant has a system failure, it nevertheless remains registered in the session. This causes a critical situation when this partner rejoins the conference.

There are a lot of multicast protocol which support at least one of the above discussed features, e. g. RMP [9], SRM [7], RMTP [8], and others, but there are only a few protocols which support all three features: the Reliable Multicast Protocol RMP and the Totem Protocol [6]. Both protocols use a token mechanism and different ordering schemes. The protocols are relatively complex and assume that larger amounts of data are continuously exchanged. In small closed video conferences signalization messages are rather rarely exchanged, only when the speaker and the group composition change. Therefore a simpler protocol can be applied.

3. GCP_{ATM}

In the next two sections we present the signalization protocol GCP (*Group Communication Protocol*) that meets the requirements discussed in the previous section. GCP was first designed for the multiparty video conference system COVIS (*Cottbus Video conference System*) which followed from the experimental video conference system GCSVA (*Group Communication and Scalability in Video conferences over ATM*) [2]. COVIS was designed for ATM. Recently a new version of GCP for IP- based networks has been designed and implemented. It is deployed in the video conference system OCTOPUS [5] and in the IP-version of COVIS which is currently being implemented. In order to distinguish the two variants of GCP we denote them GCP_{ATM} and GCP_{IP} in the sequel. In this section we introduce the basic procedures of GCP and describe shortly how they can be provided over ATM. Details are given in [4].

3.1 The video conference system COVIS

COVIS was designed to support small closed discussion groups up to 15 participants. Typical applications are project meetings or problem discussions in which several persons collaborate to discuss a design or to jointly solve a problem. Other potential applications are teleseminars in which a tutor discusses problems with students at different remote places. The main features of COVIS are a strong floor control, a dynamic scalability of video streams according to the number and the performance of the

connected hosts, and a distributed group and QoS management. More detailed descriptions of COVIS are given in [2], [3].

3.2 Protocol principles

GCP is a signalization protocol for a distributed conference management. It is used to exchange management data among the participants' systems. GCP provides an ordered, reliable, and atomic data transmission service. It further supports a dynamic group management, i.e. the participants can join and leave the conference at any time. The joining of the conference is based on an explicit invitation of the participants. The invitation procedure is not part of GCP. This can be done, for instance, by using SIP [15]. The leaving of the conference can be either explicit by the participant, called *regular LEAVE* in the sequel, or forced by the system due to a participants' failure, called *forced LEAVE*. GCP runs in parallel to the video and audio transmission protocol.

GCP was originally designed for the application over ATM AAL5. ATM applies a sender related multicast paradigm that supports closed group conferences. The design of GCP_{ATM} is based on overlaid multicast connections between the conference participants at the ATM level. Overlaid multicast optimally supports the pursued distributed management concept in contrast to a centralized multicast server or to the mapping of the multicast connections on to single duplex connections which are often used for group communication over ATM [11], [12]. It directly uses the ATM multicast capability.

For the design of GCP, we studied existing multicast protocols and concepts. A detailed discussion of these approaches cannot be included here due to lack of space. A short summary was given in Section 2. In this section we focus on the basic principles of GCP needed for the discussion in the next section. We show how the features ordered delivery, reliability, and atomicity are achieved. More details are given in [4].

GCP is a token-based multicast protocol like [6] and [9]. The participants form a logical ring on which a token rotates. Only the token holder is allowed to send messages. The other hosts have to store the messages they intend to send until they get the token. All participants have to acknowledge the reception of the messages. Unacknowledged messages are repeated. After receiving all acknowledgments the token is forwarded to the next participant. The reception of the token must be also acknowledged. If a participant has no data to send the token is immediately forwarded. GCP assures the 3 features in the following way:

Ordered delivery is given by applying the rotating token mechanism that only allows the token holder to send messages. It provides a total ordering of

the message exchange. In the original design [4] the token holder could send only one message. This appeared to be inconvenient, especially for the exchange of messages about the conference state when new participants joined the group. Now several messages are allowed.

Reliability is achieved by the acknowledgment of all messages. Each message sent by the token holder must be confirmed by the participants. This concerns all protocol procedures: the join and leave of participants, the exchange of signalization messages, and the token forwarding. Figure 1 shows this for the exchange of signalization messages. Note that all

Figure 1. Data exchange in GCP_{ATM}

messages are multicast. The participants discard the received messages if they are not needed. If a participant does not acknowledge a message it is resent up to 3 times. After that the *forced LEAVE* procedure is started which excludes the participant from the conference. This procedure is discussed below. A checksum algorithm to detect distortions of messages is not contained. It is assumed that messages are not distorted in an ATM network, but they may be lost.

Atomicity is assured by the selective retransmission of messages in combination with the *forced LEAVE* procedure. The latter excludes participants from the group who do not confirm the reception of a message or of the token after 3 requests. Thus it is guaranteed that the remaining participants received the message. The *forced LEAVE* may look very severe and unusual, but in a video conference over ATM it is very probably that a missing acknowledgement of a signaling message sent several times indicates a grave error at the participant which may also affect the transmission of the audio and video streams. In such a case it seems more appropriate to exclude the participant so that he/she can explicitly rejoin the

Figure 2. GCP_{ATM} LEAVE procedure

conference instead of setting up a complicated error recovery procedure. Figure 2 depicts the time sequence diagram of the LEAVE procedure. The *forced LEAVE* begins with multicasting a *LeaveAnnounce* message (*LA*) to all participants which is indicated to their group management modules by a *LEAVEnotification* service primitive (*LEAVEentf*) and to the excluded participant by *LEAVEconfirm* primitive (*LEAVEconf*) (if this is still possible). All participants have to confirm the exclusion by sending a *LeaveConfirm* message (*LC*) to the token holder. The *forced LEAVE* procedure is part of the *regular LEAVE* procedure called when a participant intends to leave the conference. The latter differs from the *forced LEAVE* procedure by first multicasting a *LeaveRequest* message (*LR*). The rest is identical.

Similar procedures are applied for handling token loss and duplication. To avoid token losses and duplications a 3-way-handshake is used for the token forward to the next group member. If the *TokenForward* message (*TF*) is not confirmed the forward message is repeated up to 3 times and then the *forced LEAVE* procedure for the next token holder is triggered if there is no response of that participant. Thus the rotating token also helps to detect participants' failures even if no data are transmitted, because an outstanding acknowledgment for the token leads to the exclusion of the respective participant. Token duplication can only appear if the messages of the second and third handshake are lost. This is supervised by a timer at the new token holder. If a time-out occurs the old token holder is excluded by *forced LEAVE*.

A prototype of COVIS was implemented on an experimental ATM network consisting of a FORE ASX200WG switch and five SUN workstations of different performance¹, each with an ATM interface card SBA200. In order to evaluate the performance and the applicability of GCP_{ATM} the token round trip time was measured. The measurements were carried out for system configurations with two, three, four and five workstations. In the experiments no data were sent, only the token rotated. The results are contained in Table 1. The token round trip time was too short. This led to a strong overload for the network and the involved workstations because they had too often to handle the token. Therefore, an additional delay of 10 ms was introduced in the COVIS prototype implementation (see Table 1). We extrapolated the measurements without delay to estimate the token round trip time for larger groups. It is about 40 msec for a group of 10 participants. The measurement results show that the GCP approach is applicable to small group video conferences.

Table 1 Token round trip time in GCP_{ATM}

Number of participants	Token round trip time [msec]	Token round trip time with delay [msec]
2	6,749	36,060
3	11,193	54,383
4	14,946	75,038
5	25,974	95,581

4. GCP_{IP}

The original GCP cannot be applied directly for applications in the Internet, because it assumes a sender related overlaid multicast communication infrastructure in the underlying protocol and not a point-to-point communication. IP multicast, however, uses a receiver related multicast paradigm which is not suited for closed groups. Protocols which are based on this paradigm like RMTP [8] and RMP [9] cannot be used to replace GCP although they in part support the required properties. A mapping of GCP on point-to-point connections is not acceptable, because it increases the group management and the network load for larger groups. The Small Group Multicast (SGM) protocol [10] might be a solution. It supports small closed multicast groups with a sender related multicast. The reliability of the transmission has to be assured by higher layer protocols. A shortage is that in the last segment between router and end systems it uses unicast

¹ 3 UltraSPARC1 (170 MHz, twice 140 MHz), 1 HyperSPARC (125 MHz), and 1 MicroSPARC (70 MHz).

transmission. This might be disadvantageous if there are several group members in this subnetwork. In addition, SGM requires a specialized routing software which must be available everywhere.

A software solution which can be installed in the end systems is more appropriate for this purpose. Therefore, we redesigned GCP for IP. It is applied in the video conference system OCTOPUS and for the new COVIS version over IP.

4.1 The video conference system OCTOPUS

OCTOPUS is a scalable global multiparty video conference system which was developed in a joint project between the Hong Kong University of Science and Technology and the Brandenburg University of Technology at Cottbus [5]. The objective of OCTOPUS is to set up a global video conference by connecting a number of local video conferences at different places over long distances via the Internet or ATM networks (see Figure 3). The links between the local conferences are point-to-point connections. The

Figure 3. Architecture of the OCTOPUS videoconference system (global view)

global conference is again a closed group conference. It possesses a joint floor control and a global dynamic group management. OCTOPUS does not make any assumption on the architecture of the local videoconference systems. Similarly to the principles of open systems it aims at connecting video conference systems of different structure and technology.

A unified interface to the global conference is defined by the group coordinator. It covers the possibly various local system architectures and takes over the management of the conference group. The activities comprise, among others, the group composition, the floor control, the QoS tuning, and

the delivery of the video and audio streams between the local conference systems. For example, a join or leave of a (local) participant is indicated to all participants. The local floor control strategies are replaced by a global strategy.

For the communication between the group coordinators, the GCP_{IP} is applied. RTP [14] is used to transmit the audio and video streams within the closed group. Unlike COVIS the transmission of the audio and video streams between the group coordinators is point-to-point. Not all video streams are transmitted to the remote groups. The selection depends on the decision which participants are shown at each screen and on the available bandwidth. For the selection, a voting procedure was introduced [5], [13]. The conference scenarios and the operation modes of OCTOPUS are described in [5].

4.2 Protocol principles

GCP_{IP} is a signalization protocol for systems which communicate via UDP/IP. It provides an ordered, reliable, and atomic sender related multicast service. Furthermore, it supports a dynamic group structure. The participants form a logical ring (see Figure 5Error! Reference source not found.). In contrast to GCP_{ATM} each message in GCP_{IP} are simultaneously sent in both directions: clockwise and anticlockwise. When a participant receives both messages it can be assumed that the ring is not broken and that the other group members also received these messages. The reception of the message has to be confirmed the same way. In case of a broken ring the token holder repeats the message up to 3 times. To determine the gap in the ring the third message must be confirmed by all participants passed. Then the *forced LEAVE* procedure is started to exclude the nonactive participant. The ring is set up during the initialization phase. A new participant is included into the ring in such a manner that his/her neighbours are situated, if possible, in the same network segment. This avoids long token rotation times.

GCP_{IP} assures the three basic features of GCP as follows. **Ordered delivery** is forced by the token mechanism which only allows the token holder to send signalization messages. The other participants only confirm the reception of the messages. **Reliability** is achieved by the double logical ring which connects the group members. It guarantees that all messages are delivered to all active group members. **Atomicity** is ensured by the *forced LEAVE* mechanism which as in GCP_{ATM} excludes nonactive group members to reestablish the ring.

In the sequel we describe the protocol phases of GCP_{IP} in more detail.

SETTING UP A CONFERENCE GROUP

When a conference group is supposed to set up the initiator first establishes a connection with one conference partner using a 3-way-handshake. The initiator (in the OCTOPUS system this is the group coordinator) delivers a *JOINreq* primitive to its GCP_{IP} entity which sends a *JoinRequest* message (*JR*) to the selected partner. There the request is indicated by a *JOINind* primitive. The invited partner can accept or reject the invitation with a *JOINresp* or a *JOINrej* primitive, respectively. The answer is sent back to the initiator by a *JoinConfirm* (*JC*) or *JoinDisconnect* (*JD*) message. When the session is rejected the initiator is informed by a *JOINdis* primitive. The acceptance of the invitation is confirmed by a *ConnectAddition* message (*CA*) by the GCP_{IP} entity. Both participants are

Figure 4. Invitation of a new member

informed about the successful connection by a *JOINntf* primitive. Then the initiator creates a token and forwards it to the partner. Further group members are invited as new participants as described next.

INVITATION OF A NEW PARTICIPANT

Each group member is allowed to invite a new participant upon an oral agreement with the other conference partners. The invitation is triggered by a *JOINreq* and follows the same procedure as the conference initialization (see Figure 4). After receiving the confirmation the inviting group member determines between which partners the newcomer is included into the ring to optimize the token round trip time. The group members are informed by a *ConnectionAddition* (*CA*) message about the position of the new participant.

Each group member proves whether the new partner is its neighbour. In this case they update their forward address and forward the *CA* message to the new partner. The *CA* also contains state information of the ring for the new partner. The new partner forwards the message to its new neighbour. The same procedure is executed for the other direction. In Figure 4 we distinguish the *CA* messages sent in opposite directions by adding *L* and *R* for left and right, respectively. The successful join of the new group member is indicated by a *JOINntf* to the application.

DATA TRANSMISSION PHASE

The data transmission in GCP_{IP} corresponds to that of GCP_{ATM} as described in Section 3.2. The difference is that the *DT* messages are also sent in both directions of the logical ring. Figure 5 represents the principle. The Figures 5a) and 5b) show the forwarding in both directions. In Figure 5c) the group members received the *DT* message and indicate this to their application by a *DATind* primitive. The successful transmission is implicitly

Figure 5. Data Transmission of GCP_{IP}

confirmed when the both messages pass every participant. An unsuccessful transmission is repeated up to 3 times. After that the *forced LEAVE* procedure is called. In Figure 5d) the token is forwarded to the next group member.

LEAVING THE CONFERENCE

GCP_{IP} applies the *regular* and the *forced LEAVE* procedures as GCP_{ATM} . Figure 6 depicts the time sequence diagram for this procedure. In case of a *regular LEAVE* the leaving group member informs their neighbours by an *LR* message. The neighbours confirm this by a *LC* message and change the

Figure 6. Leaving of a participant

addresses of their respective neighbours. The leaving member indicates the leave to the application by a *LEAVEntf*. The rest of the procedure concerns only the remaining group members. This is also the starting point for the *forced LEAVE* procedure. The both neighbours of the left or excluded member inform all group members by sending *LA* messages in both directions. This has to be confirmed by an *LC* message. After having received both *LC* messages each group member sends a *LEAVEntf* primitive to its application to indicate the leave. If there remains only one group member the conference is terminated.

4.3 Performance evaluation

To evaluate the performance of the protocol we measured the round trip time of the token and the average delay of data packets between sending and delivery. The round trip time indicates how fast requests of the application to the group management are handled. The delay is a measure for the distribution time of the signalization messages. The sum of both times is the maximum reaction time of the system to group events. The assumption of a simple addition can be made here, because the frequency of the data exchange is relatively low. The measurement results are depicted in Figure 7.

The measurements were carried out on a 100 Mbps Ethernet with an average daily load. The involved hosts were SUN UltraSparc 10, SUN SparcStation 20, and SUN SpacStation LX workstations. As expected the round trip time and the delay increased quasi linear. The mostly slightly higher delay of the data packets can be explained by the double handling of the packets for both directions. For 8 participants, the maximum reaction time is about 210 msec and approximately for the planned upper size of 15

Figure 7. Token Round-Trip-Time and average packet delay

participants about 360 msec. The estimated lost of bandwidth due to the rotating token mechanism is approximately 3 per cent of the average bandwidth needed for a video conference with 15 participants. The measurements prove that GCP_{IP} can be deployed for the planned purpose, since the delay of events such as speaker change or participant's join and leave are very low. For small groups, GCP_{IP} represents an alternative to other protocols in the Internet.

For applications in global networks like the OCTOPUS project, the transmission delay has to be taken into account and added to the above measured values. The round trip time for transmissions between Cottbus and Hong Kong fluctuate between 350 and 500 msec. This is still a remarkable overhead. Closed group video conference meetings in their majority, however, will take place in much smaller regions (e.g. university campuses, cities, states, countries). GCP_{IP} offers an applicable solution for this.

5. FINAL REMARKS

In the paper we have presented the group communication protocol GCP for supporting a distributed management of closed video conferences. GCP aims at small closed groups. It provides an ordered, reliable, and atomic data delivery services which ensures the consistency of the management data at the participants. GCP exists in two variants. GCP_{ATM} has been successfully applied for video conferences over ATM. In this paper we have shown with how the features of GCP can be also provided in IP networks. The measurements show that GCP_{IP} is applicable for supporting closed groups meetings in the Internet. GCP can also be applied to other areas like joint editing which require a floor control.

In our current research we deploy both GCP variants. We are reimplementing COVIS on the German scientific research network G-WiN that provides high bandwidth up to 2,5 Gbps without guaranteed QoS. This requires changes in the COVIS_{IP} architecture which affect beside the video stream scaling also the group communication. The architecture of COVIS_{IP} is similar to that of OCTOPUS. It replaces the group coordinators by video gateways. GCP_{IP} is used for the group management.

REFERENCES

- [1] Handley, M.; Crowcroft, J.; Bormann, C.; Ott, J.: Very large conferences on the Internet: the Internet multimedia conferencing architecture. *Computer Networks* 31 (1999)
- [2] Beier, I.; Koenig, H.: GCSVA - A Multiparty Videoconferencing System with Distributed Group and QoS Management. *Proc. of the 7th International IEEE Conference on Computer Communications and Networks (IC³N'98)*, Lafayette, USA, 1998
- [3] Beier, I.; Koenig, H.: Distributed QoS Management for a Multiparty Videoconferencing System. In: Yongchareon, T.; Agesen, F. A.; Wuwongse, V. (eds.): *Intelligence in Networks. IFIP TC6 WG6.7 5th International Conference, SMARTNET'99*, Pathumthani, Thailand, Kluwer Academic Publishers, 1999
- [4] Beier, I.; Koenig, H.: A Protocol Supporting Distributed Group and QoS Management. In: *Proc. IEEE Conference on Protocols for Multimedia Systems - Multimedia Networking (PROMS-MmNet'97)*. Santiago, Chile, 1997
- [5] Chanson, S. T.; Hui, A.; Siu, E.; Beier, I.; Koenig, H.; Zuehlke, M.: OCTOPUS - A Scalable Global Multiparty Video Conferencing System. In: Dixit, S.; Somani, A.; Park, E. K. (eds.): *Proc. of the 8th International Conference on Computer Communications and Networks (IC³N'99)*, Boston, Massachusetts, USA, IEEE Operations Center, 1999
- [6] Agarwal, D.A.: Totem: A Reliable Ordered Delivery Protocol for Interconnected Local-Area Networks, Ph.D Thesis, University of Santa Barbara, Dec 1994
- [7] Floyd, S.; Jacobsen, V.; McCanne, S.; Liu, C.-G.; Zhang, L.: A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing, *Proceedings of the ACM SIGCOMM'95*, Cambridge, USA
- [8] Lin, J.C.; Paul, S.: RMTP: A Reliable Multicast Transport Protocol, *IEEE INFOCOM'96*, San Francisco, March 1996
- [9] Whetten, B.; Montgomery, T.; Kaplan, S.: A High Performance Totally Ordered Multicast Protocol, *IEEE INFOCOM'95*
- [10] Boivie, R.; Feldmann, N.: SGM - Small Group Multicast. Internet Draft Proposal, July 2000

- [11] Chan, D. L. S.; Chanson S. T.: Scalability Support for Multiparty Multimedia Communications. In: ACM/Springer Multimedia Systems Journal, 6 (1998) 2
- [12] Gauthier, E.; Le Boudec, J.-Y.; Oechslin, P.: SMART: A Many-to-Many Multicast Protocol for ATM, IEEE Journal on Selected Areas in Communications, 15 (1997) 3
- [13] Koch, H.: A solution of the visibility problem for the video conference system OCTOPUS. Master thesis, BTU Cottbus, Department of Computer Science, 2000.(in German)
- [14] Schulzrinne H.; Casner S.; Frederick R.; Jacobson V.: RTP: A Transport Protocol for Real-Time Applications. RFC 1889, 1996
- [15] Handley, M.; Schulzrinne, H.; Schoolre, E.; Rosenberg, J.: SIP: Session Invitation Protocol. RFC 2543, March 1999