

GCAP: A New Multimedia Multicast Architecture for QoS

Michel Diaz^{1*}, Roberto Canonico², Luis Costa³, Serge Fdida³, David Hutchison⁴,
Laurent Mathy⁴, Andreas Meissner⁵, Stephane Owezarski¹, Rolland Vida³, Lars Wolf^{5,6}

¹ LAAS-CNRS, 7 Avenue du Colonel Roche, 31077 Toulouse cedex 4, France
{diaz, sowe}@laas.fr

² Università di Napoli Federico II, Dip. di Informatica, Via Claudio 21, 80125 Napoli, Italy
roberto.canonico@unina.it

³ LIP6 - University of Pierre et Marie Curie, 4, place Jussieu, 75252, Paris Cedex 05, France
{Luis.Costa, Serge.Fdida, Rolland.Vida}@lip6.fr

⁴ Lancaster University, Computing Department, Lancaster LA1 4YR, UK
{dh, laurent}@comp.lancs.ac.uk

⁵ Fraunhofer IPSI, Dolivostrasse 15, 64293 Darmstadt, Germany
Andreas.Meissner@ipsi.fraunhofer.de

⁶ Technical University of Karlsruhe, Zirkel 2, 76128 Karlsruhe, Germany
Lars.Wolf@uni-karlsruhe.de

Abstract. Despite its obvious suitability for distributed multimedia applications, multicasting has not yet found widespread application. Having analyzed shortcomings of today's approaches, we devise in the *GCAP project* a new end-to-end transport architecture for multimedia multicasting that supports partial order and partial reliability. In this paper, we argue that, at the network layer, *single-source multicasting* (PIM-SSM) should be chosen. Consequently, our *Monomedia Multicast* protocol provides, along with reliability and QoS monitoring functionality, an ALM based multicast solution referred to as TBCP (*Tree Building Control Protocol*), to be used as back channel for SSM, e.g. for retransmission requests. On top of the Monomedia protocol, our *Multimedia Multicast* protocol handles multimedia *sessions* composed of multiple monomedia connections: The FFTP (*Fully Programmable Transport Protocol*) allows applications to specify, through its API, the (global) synchronization and (individual) reliability requirements within a multimedia session. Our group management approach is focused on *group integrity*.

1 Introduction

Distributed *multimedia* applications typically require Quality of Service (QoS) support from networks and systems due to their handling of time-critical media such as audio and video. Additionally, in many of such applications, several participants are involved, which makes the use of *multicast* transmission methods attractive in order to save network bandwidth. Unfortunately, due to the management complexity of current multicast protocols, they have not found as widespread a use as expected. The goal of our work within the framework of the *GCAP project* [8] is to address these shortcomings and to provide for QoS support for applications [6], to simplify

* Corresponding author. Further authors are listed in alphabetical order.

the development of applications, and to make usable and manageable multicast support available by introducing a *simplified multicast model*.

To address the requirements of multimedia multicast applications, we provide mechanisms to handle and coordinate a set of receivers and a set of time-critical multimedia objects exchanged between many communicating entities. Communication protocols at the transport and higher layers are necessary, e.g. for the transmission of data between the communicating end systems, but also for higher-level coordination functions like group management and integrity (for instance, to ensure that all mandatory group members of a videoconference are available and, hence, group integrity requirements are fulfilled). In today's approaches, such a design and implementation effort is part of the development at the application layer – placing this burden again and again on every application developer and wasting development time, as opposed to an approach where such functionality is provided once and can be reused. Hence, GCAP follows a general approach and proposes the design of new protocols within a new *transport architecture*. Furthermore, it aims to rapidly deploy and to optimize such innovative architectures using an *active network* based approach.

To support multicast, multimedia transport services and protocols for group communication, GCAP adds new functions to the present Internet architecture. There are two ways for realizing this in the IP protocol stack:

- 1) to design a whole new end-to-end transport protocol to extend the classical UDP/TCP, and to use it as the new end-to-end transport protocol;
- 2) to introduce, on top of UDP and TCP, a new layer (referred to as the *GCAP transport layer*), to be located between the traditional transport layer and the application layer, to handle the new functions proposed for GCAP.

For ease of implementation and distribution, we have opted to use the latter.

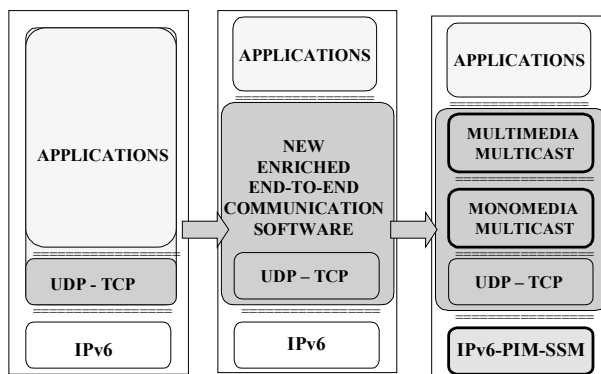


Figure 1. From UDP/TCP to the GCAP Architecture

The basis of our architecture is the PIM-SSM IPv6 layer where a flow of data is sent from one source to a set of receivers. This set of flows in the networks is called a multicast *channel*. Of course, based on IP, the channel handles datagrammes, without reliability. The second step of GCAP is to provide in a generic way the *reliability* required by different applications. For this reason, the new GCAP transport layer, dedicated to support multimedia multicast applications, has been designed. It consists of two sub-layers, *monomedia multicast* and *multimedia multicast*, able to provide

users with a complete set of reliability services. In the first sub-layer, this reliability is designed as a *monomedia connection*, and in the second as a *multimedia connection*, i.e. a set of monomedia connections, where synchronisation and losses are related to provide a general *partially reliable* service. This set of monomedia connections is also called a *multimedia session*. The resulting global architecture is shown in figure 1.

The outline of this paper is as follows. Single source multicasting is discussed in section 2. Monomedia and multimedia aspects of our work are detailed in sections 3 and 4, respectively. Finally, section 5 concludes this paper.

2 Multicast Network Support

This section explains the multicast network support choice made in the GCAP project. The first part briefly reviews the current IP Multicast architecture to understand why its deployment in the Internet was slowed down. The second part presents the source-specific multicast service model which was chosen as the network support in the GCAP architecture.

2.1 The current IP Multicast Architecture

The current IP Multicast architecture is composed of a service model that defines a group as an open conversation from m sources to n receivers, an addressing scheme based on IP class-D addresses, and routing protocols. Any host can send to a multicast group and any host can join the group and receive data. A multicast group is identified by a class-D IP address which is not related to any topological information, as opposed to the hierarchical unicast addressing model. Therefore, multicast address allocation is complicated and multicast forwarding state is difficult to aggregate.

At this time, there is no scalable solution to inter-domain multicast routing in the Internet. The currently adopted architecture is based on the IGMP/PIM-SM/MBGP/MSDP protocols. Hosts report their interest in specific multicast groups to their local router through the IGMP protocol (*Internet Group Management Protocol*) [5]. Intra-domain routing is done by PIM-SM (*Protocol Independent Multicast – Sparse Mode*) [7]. PIM-SM domains are connected using the MBGP (*Multiprotocol Extensions for Border Gateway Protocol - 4*) [2] and MSDP (*Multicast Source Discovery Protocol*) [13] protocols.

The PIM-SM protocol is adapted to sparsely populated groups for which flood-and-prune techniques are inadequate. PIM-SM builds unidirectional shared trees that are rooted on a special router, called *rendezvous point* (RP). Inside a PIM-SM domain each multicast group is mapped to one RP. A multicast source encapsulates the data in unicast and sends it to the RP. The RP then multicasts the data on the shared tree.

If the sending rate of the source exceeds a previously fixed threshold, PIM-SM builds a source tree for this specific source. Current implementations set the used threshold to one packet. As soon as the first packet is received, the protocol changes the shared tree for the source based one. The RP and the shared tree are therefore used only for source discovery.

The use of PIM-SM in the inter-domain level has two main problems: designing a scalable mechanism for mapping multicast groups to RPs and the fact that ISPs do not desire to host the RPs for other ISPs' multicast groups. Besides, since PIM-SM relies on the unicast routing protocol to construct multicast trees, join messages may reach non-multicast routers, complicating PIM's operation. The short-term solution to these problems is to use MBGP and MSDP. MBGP allows multiple routes to be announced for different protocols. In this way, routers may construct one routing table with unicast-capable routes and another with multicast-capable routes. MSDP provides a solution to the ISP interdependence problem. RPs within one domain are interconnected and connected to RPs in other domains using MSDP to form a loose mesh. When a source in a specific domain starts sending, the RP in this domain sends a MSDP *Source Active* message to RPs in other domains. Members located in other domains send source-specific join messages following the MBGP routes in the inter-domain level. This solution does not scale because RPs exchange information about all active sources located at all domains.

The complexity of the IP Multicast architecture slowed down the multicast service deployment, to the benefit of application level solutions or distributed caching architectures. A different approach to this problem is to simplify the architecture. This is the main idea of the Source-Specific Multicast (SSM) service model.

2.2 The Source-Specific Multicast Service

The SSM model [10] restricts the multicast distribution to *1 to m* introducing the multicast channel abstraction. A distribution tree is constructed for each channel. A channel is identified by a (S,G) pair, where S is the unicast address of the source and G is a class-D multicast address. Only S is able to send data to channel (S,G), channels (S1,G) and (S2,G) being different by definition. Address collision problems are avoided as the unicast address of the source is unique. Multicast address allocation becomes a problem local to the source. The multicast address G does not need to be globally unique anymore.

In IPv4 an exclusive address range, 232/8, has been allocated by IANA for SSM channels, enabling the coexistence of source specific service with the traditional IP Multicast (any-source multicast) service. In IPv6, the range FF2::/11 through FF3x::/11 is defined for SSM services [9].

Multicast channel subscription in IPv4 is done using the IGMPv3 [5] protocol. Version 3 of IGMP adds support for "source filtering", that is, the ability for a host to report interest in receiving packets *only* from specific source addresses, or from *all but* specific source addresses, sent to a particular multicast address. This information may be used by multicast routing protocols to avoid delivering multicast packets from specific sources to networks where there are no interested receivers. IGMPv3 provides a superset of the functionality needed by SSM, namely, to report the interest on a (S,G) pair.

In IPv6, IGMP was renamed as MLD (*Multicast Listener Discovery*) and included in ICMP (*Internet Control Message Protocol*). Version 2 of MLD [17] implements the same functionality as IGMPv3, in particular, the ability to report source filters for each multicast address listened to.

The other component of the SSM model is the PIM-SSM [3] routing protocol that constructs the multicast distribution tree. PIM-SSM is a modified version of PIM-SM that implements the source-specific service on the reserved SSM address range. PIM-SSM is able to construct source-specific trees with no need to previously construct a shared tree (as PIM-SM does). PIM-SSM has therefore no need for RPs and all the mechanisms for RP election and group mapping. On the other hand, PIM-SSM supposes that receivers know the channel identifier, i.e., the address of the source and the class-D multicast address. The way channel identifiers are discovered is not defined by the SSM service, but it could be done by mechanisms such as e-mail, web announcements, the *sdr* tool, etc.

When a receiver wants to join a multicast channel, it sends an IGMPv3 (IPv4) or MLDv2 (IPv6) request to its designated router, specifying the (S,G) address pair. If the address G is in the exclusive address range, the designated router sends a *join(S,G)* message towards the source in order to be added to the distribution tree of channel (S,G). If instead the address G is outside the SSM address range, the router behaves according to the PIM-SM specification, sending a *join(*,G)* request towards the RP.

3 Monomedia Multicasting

In this section, we address the *monomedia multicast sub-layer* in GCAP, whose role is to provide meaningful end-to-end services, on top of the PIM-SSM model. However, at this level of the architecture, the focus is on the control of individual communications, rather than on their integration and interpretation as part of a multimedia session. Hence, this monomedia multicast transport sub-layer is mainly concerned with mechanisms for the control and monitoring of data sent over a PIM-SSM channel. The services provided by this layer enhance the PIM-SSM service with several forms of reliability and monitoring.

As already mentioned, one of the main characteristics of the SSM model is that only the SSM channel source is allowed to send on the channel. This property has an important implication for higher level protocols and applications: no multicast “back-channel” is available for receivers to multicast feedback to the group. This lack of multicast facility to send control/feedback information, together with the restriction of a single, well-known sender, breaks several transport and application level protocols.

It is worth noting that, although a class of protocol mechanisms relying on statistical methods and based solely on direct feedback to the source have been proposed in the literature [4] [14], these mechanisms only perform efficiently for large multicast groups (exceeding 10^4 participants) and are therefore not applicable with the vast majority of interactive applications considered in GCAP. We have thus opted for an approach using transport-level distribution structures for control purposes.

3.1 The Tree Building Control Protocol

By transport-level distribution structure, we mean that the GCAP monomedia multicast sub-layer builds, and maintains, its own “communication overlay”,

encompassing all relevant transport entities, using readily available unicast communications. This transport-level distribution structure is built as a spanning tree connecting the source and the receivers of the associated PIM-SSM channel.

Since the solution of sending all control information to the SSM channel source and then letting the channel source “reflect” this information back onto the SSM channel can clearly not be scalable, a different solution based on application level multicasting has been devised in GCAP. This approach requires the building of a bi-directional spanning tree by means of an application level tree building protocol, called TBCP. This *Tree Building Control Protocol* [18] is a new tree building protocol designed at Lancaster University. TBCP can build a loop-free spanning tree connecting TBCP entities that may run either in end systems or in programmable *edge devices*. Such a spanning tree is built by means of a distributed algorithm that does not require interaction with multicast routers or any knowledge of network topology.

TBCP has been designed according to the *Application Level Multicast (ALM)* model, also known as application-based distribution. ALM is a new technique used to provide multicast distribution services in situations where no support is provided by the network for multicast. This situation occurs, for instance, when the network infrastructure does not support the IP-multicast routing protocols. As we already mentioned in the previous section, another case in which the ALM approach may be helpful is when the network only supports an SSM multicast service, and the user wants to deploy end-to-end communication protocols that require the availability of a full duplex multicast distribution tree.

When the ALM technique is used to implement multicast communication, each end system that participates in the multicast group is also responsible for forwarding received data to all other ALM entities connected to it in the tree. The advantage of performing data forwarding at the application level is that each end system may be programmed to either passively retransmit every single piece of data it receives or actively filter this information according to some protocol-specific logic (for instance, upstream traffic may be aggregated to avoid the source implosion problem, while downstream traffic may be filtered to adapt to the link congestion or the actual computing power of downstream receivers). The tree built by TBCP is used in GCAP to deliver control data among the members of a multicast session. By using a TBCP tree connecting transport-layer entities (which may run either in end-systems or in the active edge devices discussed later), it is possible to implement the control protocols that require a multicast distribution of messages originated by SSM receivers.

In the process of building the TBCP tree, TBCP entities interact through a sort of signalling protocol which uses point-to-point TCP connections to exchange protocol messages. To join a tree, a TBCP entity needs to establish a TCP connection with the root TBCP entity, and begin a join procedure (whose details are outside the scope of this paper). Hence, a TBCP tree may be identified by the pair (S,SP), where S is the IP address of the node on which the root entity is running, and SP is the TCP port number used by the root entity to listen to new connection requests.

Once a TBCP tree has been built, it can be used to deliver control data among all the nodes that have joined the tree. For a TBCP tree to be used by several protocol entities, a set of *data channels* is associated to a TBCP tree. Each data channel is uniquely identified by a 16-bit identifier, called *ChannelID*. Figure 2 shows a TBCP tree with three active data channels.

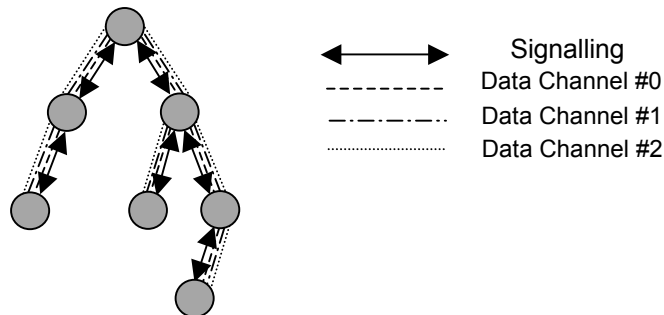


Figure 2. A TBCP Tree with three active Data Channels

Finally, it should be noted that the term *data channel* used by TBCP must not be confused with the term *SSM channel*. Incidentally, the way information is delivered on a specific TBCP Data Channel depends on how the forwarding entity for that channel is programmed. For a given tree, in fact, some data channels may be used as broadcast channels (i.e. a datagram sent by any node is forwarded to all the other tree nodes), while other channels may implement an SSM behaviour (i.e. only the tree root is authorised to inject packets into the tree). More complex usage of a data channel may be requested by other protocols if some form of message aggregation is required.

3.2 Reliability

Reliability in the Monomedia Multicast service sub-layer is realized by reliable multicast protocols running on top of the TBCP tree. Such protocols will be designed as a "mapping" of well known reliable multicast protocols (e.g. RMTP, PGM, etc.) onto the GCAP architecture, i.e. these well known reliable multicast protocols will be modified to fit GCAP communication constraints.

Three reliable multicast services will be supported:

- *Fully unreliable service.* This service provides best-effort datagram delivery where no provision is made to enable subsequent attempts to recover lost data.
- *Unreliable with selective recovery requests service.* This service provides best-effort datagram delivery but makes necessary provisions in order to enable the recovery of lost data for a specified recovery period of time. It should be noted that this service is "reactive" in the sense that the recovery period as well as the recovery requests are specified by the service user.
- *Fully reliable service.* This service provides a loss free transfer facility.

3.3 Monitoring

The *QoS monitoring* on a monomedia multicast connection will be based on an adaptation of the RTP/RTCP protocol. Here the situation is different than with the other protocols considered so far: with RTCP, low frequency feedback has to be distributed to all the participants (including the source) of the communication.

Because this feedback is kept to a small percentage of the overall data traffic, and each participant should see the feedback messages "as is", the overhead of distributing such feedback over the TBCP tree seems excessive: RTCP follows a model where sending the feedback, in unicast, to the SSM source, and having the source "reflect" it to the group on the SSM tree, is well suited.

Another useful service the Monomedia Multicast sub-layer provides is basic group *integrity condition checking*, defining minimal operating conditions on the monomedia multicast connections, such as the minimum number of participants or mandatory participants. These integrity mechanisms are well suited to run on top of the TBCP tree and form a basis for the more comprehensive integrity framework discussed in section 4.2.

3.4 Edge Devices

One of the most important properties of multicast is scalability. As a consequence, following the classical approach and ensuring compatibility with the TBCP monomedia sub-layer, our multicast architecture has been designed as a tree, where a node has to handle the reliability of its children nodes. The first implementation we are developing only has the sender, one intermediate tree level and the receivers. For simplicity and also because this is quite in agreement with real networks, the intermediate level is located at the entrance of the edge LANs where the users are, hence it appears in a specific router called an *edge device*, as it represents the connection of the edge of the LAN with the core network.

4 Multimedia Multicasting

In this section, we discuss how GCAP supports the requirements of multimedia applications. To address these requirements, GCAP is targeting a new transport layer where the relations between the different components of multimedia communications are defined at an extended end-to-end interface, i.e. a general *multimedia multicast API*. This allows the specification of powerful QoS parameters and of global multimedia synchronization requirements. In addition to these transport layer issues, we discuss how *channel* and *session integrity* is defined and maintained in GCAP.

4.1 Partial Order and Partial Reliability Transport

The multimedia multicast sub-layer, based on the monomedia multicast sub-layer, has to handle a set of multimedia *sessions*, each session consisting of multiple monomedia *connections*, which requires to specify the needed relations and interdependencies between these connections. Moreover, in several communication scenarios for group communication, it is also important and needed to maintain the integrity of the multimedia information over the lifetime of the session.

As a general concept, GCAP uses a partially reliable and partially ordered model to define a session. More precisely, the reliability and synchronization requirements

characterize this session. First, for *partial reliability* support, the model is based on the consideration that a fully reliable communication increases the communication delay which may be worse than tolerating some multimedia losses, because losing video frames can be acceptable in given applications. Second, for *partial order* support, the model is based on the consideration that a fully ordered communication does not represent general multimedia objects, where the different sub-objects are defined in parallel and in sequence, and are then synchronized between themselves. Using such a well-defined specification, for instance using Time Stream Petri nets, an application can specify which packets must be received to maintain a given QoS. The model is used to represent the partial order, i.e. the synchronization, and the partial reliability, i.e. the number of requested receptions, of the media sub-object of a multimedia presentation. We point out that:

- 1) TCP only provides a fully ordered and a fully reliable service, while UDP provides only an unordered and unreliable service, but
- 2) multimedia data streams require both:
 - partial reliability (for example, it is not troublesome if some images of a 25 images/s video stream are lost), and
 - partial order (multimedia objects as serial and/or parallel compositions of multimedia data).

As a solution, a partial order connection protocol has been proposed [1] [15]. Its extension in GCAP has led to a new transport protocol, *FPTP (Fully Programmable Transport Protocol)*[†]. It delivers data units sent on several connections, following a given partial order. UDP and TCP can be seen as two specific cases of FPTP, and the partial order is any order between the total order (as TCP) and no order (as UDP).

Regarding partial reliability, when an acceptable loss of a message is detected:

- 1) the missing object can instantaneously be declared lost (leading to earliest indication of losses), and
- 2) the data received (in general the one that leads to the detection of the loss) can be delivered immediately to the application (leading to earliest delivery).

Of course, if the loss cannot be accepted in terms of requested reliability, recovery and retransmission will occur.

This required partial reliability, defined on top of the partial order model, results in earliest loss indications and deliveries; it is deduced from the application requirements [16]. Partial reliability is managed by *groups of media*, where the receiving entity can directly handle the set of all streams, and so it is possible to declare losses on all streams of the same multimedia connection. Handling the losses is of course derived from the synchronization model of the multimedia object.

With regard to partial order, what must be clearly understood is that the multimedia sub-layer handles together, as a group of connections, the synchronization of the different media. The basic Petri net model defines the partial order as the synchronization between the different objects of the multimedia presentation. An

[†] The detailed FPTP API will be provided and made available for comments and improvements near the end of the project, in November 2001, in a separate and dedicated public report to be found at [8]. Another public report will give the detailed GCAP measurements obtained by using the European infrastructure, i.e. the national research networks *Renater*, *Rederis*, *G-WiN*, *Super-Janet*, *ACOnet*, and their European interconnection *TEN155*.

example is given in figure 3. The higher part represents the partial order and the partial reliability of a simple application, i.e. a 16 unit period of a continuous object.

We now discuss in more detail the multipoint architecture. As the FPTP is a protocol based on the monomedia multicast infrastructure, it has to be based on the underlying spanning tree. The first prototype of the GCAP multimedia multicast sub-layer provides its services, integrating the concepts of partial order and partial reliability, on top of a fully *unreliable* monomedia multicast service.

During a session, the source sends data using PIM-SSM on the several monomedia data flows that have been created. This data is received by the end user. If a QoS violation is detected, the receiver asks the source to send the missing data via the monomedia TBCP channel interface. On the contrary, data fulfilling partial reliability is transmitted to the application by respecting the multimedia partial order synchronization specification and model. If some losses are detected, and if the QoS is violated, retransmission is requested.

It is important to understand in multicast that, for a set of users, each user can have his own definition of partial reliability. Of course, all of them use the same synchronization model as applicable to the multimedia presentation.

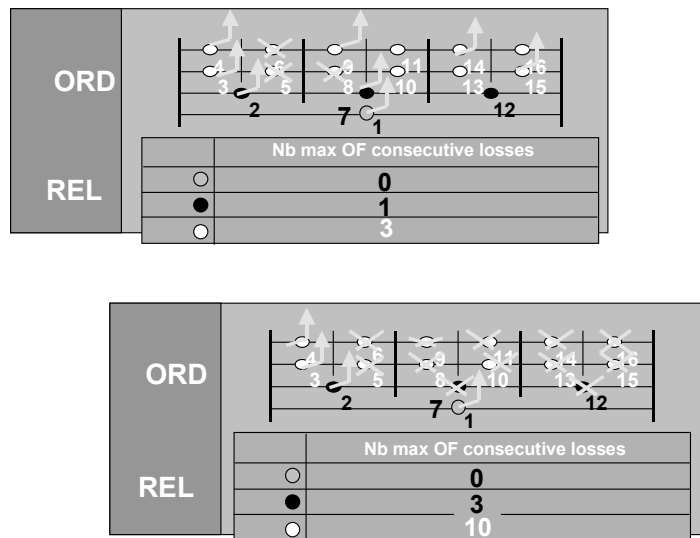


Figure 3. Synchronisation and Partial Reliability for two Users

Figure 3 gives an example that shows two users requiring two different (partial) *reliability* values, i.e. giving two different sets of numbers of *acceptable losses* in the connection. The upper part of the figure is related to user 1 and the bottom part to user 2. It can be seen that the *synchronization* between the different media is the same for both users, as the multimedia object being sent by the sender is the same for both.

The difference between the two users appears in terms of acceptable losses. Here this means, for instance, that user 1 accepts, for the upper two of the four flows, (up to) three losses in the synchronization period given in the figure, while user 2 accepts (up to) ten losses in the same two flows during the same period. Moreover, the partial

reliability, and thus the quality, is higher for the third medium of user 1 (one for user 1 and three for user 2, for the third flow) while it is equal for the fourth flow (zero for both). Of course, there is a direct relationship between the number of acceptable losses and the available throughput. A user having a high speed link, like user 1, can, for example, request 20 images per second, while another user using a slow link is able to request only one image per second, i.e. a quality ten times lower than that of user 1.

The behavior of the receiving entity of the protocol is then simply as follows: ACKs are generated as long as acceptable losses occur; and if too many messages are lost, i.e. when the number of the actual losses becomes higher than the number of acceptable losses, then error recovery starts.

Note that the interest of the approach is twofold:

- First, the definition and the specification of the partial reliability is local and so easy to define, as the user should know the capability of his network; and
- Second, the handling of the error acceptance and of the error recovery is also local, i.e. it can easily be different for each of the users of the multicast.

These two properties allow the system to be optimized from the user point of view, with the user having the poorest connection being able to request and handle the poorest quality without degrading the quality of the other participants of the multicast.

Finally, because the FFTP protocol has to have a recursive behavior for scalability, we decided to deploy it between the edge devices located at the entry of the LANs.

4.2 Channel and Session Integrity

While our FFTP is tailored to specific multimedia multicast *data transmission* requirements, a second multimedia multicast aspect addressed by GCAP is *multicast group integrity* (not to be confused with the *integrity of the multimedia information* discussed in the previous section). This concept, as first proposed in [11], refers to *conditions* imposed on groups with regard to membership set, member roles, topology, and group organization. Associated *action policies* state how to re-establish integrity in case it has been found violated. In GCAP's multimedia multicasting context, we apply an adapted version of the framework in [12] to *SSM channels* and multimedia *sessions*.

Two entities are involved in integrity control: A central *session manager* M, also responsible for each *channel's* integrity, and a *controller module* at each user U, who obtains permission from M for any operation relevant to session or channel integrity management. We refer to *users*, identified by a user ID, instead of *hosts*; a current mapping is kept by M. A user U who wants to join one or more SSM channels first requests permission from M for his intended operations, and, if granted, performs MLDv2 and TBCP join operations by way of his controller. Any subsequent member status changes, such as leaves, also require permission by M. Only privileged users, as identified according to a *user directory*, may modify integrity conditions and action policies, or establish or delete SSM channels within the session. All channel members are, implicitly, session members, too.

M maintains, per *channel*, a list of members with their roles, as well as a set of integrity conditions against which requests are checked (such as mandatory members

and non-admissible users, or the minimum and maximum number of members permitted) and action policies (such as what to do if a mandatory member has died), complemented by *session* integrity conditions and action policies. For sessions, we allow *topological* restrictions by explicitly allowing or denying members in certain domains. Voting among channel or session members is supported by M as a means of reaching a group management decision, e.g. on a join request by a new user. Users may at any time obtain from M a set of applicable integrity conditions in order to avoid useless requests, such as attempting to join only the video channel while session integrity conditions mandate to join both the audio *and* the video channel.

5 Conclusion

In this paper, we have described the new approach to multimedia multicasting we are developing in the GCAP project. On top of PIM-SSM at the network layer, our monomedia multicast protocol provides a multicast feedback channel by way of a TBCP control tree. The special requirements of multimedia applications, namely partial order and partial reliability, are supported in our new FFTP that, by offering a powerful multimedia multicast API to applications, overcomes the limitations of current application layer multicasting approaches. Furthermore, group integrity was discussed in our multimedia multicasting context.

Having successfully tested TBCP at the monomedia layer as a building block, we are now preparing two major experiments involving the GCAP multimedia protocol: a media server experiment with time-sensitive data, and, involving the entire GCAP protocol suite, a multi-party multicast video conference experiment stretching across the national research networks of our countries.

We believe that GCAP is setting the stage for multimedia multicast research. However, a set of work has still to be done to obtain optimised software for handling multimedia and multicast, including models, simulations, congestion control, and to adapt it to real multi-network environments including fiber, wireless and satellite networks.

Acknowledgements

The GCAP work [8] has been supported by the European Union within the Information Societies Technology (IST) Framework as Project IST-1999-10504. Contributors are: Centre National de la Recherche Scientifique LAAS (France), Telebit Communication (Denmark), 6WIND (France), Fraunhofer Institute for Integrated Publication and Information Systems - FhG IPSI (Germany) (formerly GMD – Forschungszentrum Informationstechnik), Université Pierre & Marie Curie – Laboratoire LIP6 (France), University of Lancaster (UK), Universidad Carlos III de Madrid (Spain), Alcatel Space Industries (France), Telekom Austria (Austria), Technical University of Karlsruhe (Germany), and Ecole Nationale Supérieure Ingénieurs En Constructions Aéronautiques (France). The writing of this paper was coordinated by Andreas Meissner of Fraunhofer IPSI.

References

1. P. D. Amer, C. Chassot, T. Connolly, P. Conrad, M. Diaz: Partial order transport service for multimedia and other applications. *IEEE/ACM Transactions on Networking*, October 1994, Vol. 2, No. 5
2. T. Bates, R. Chandra, D. Katz, Y. Rekhter: Multiprotocol Extensions for BGP-4, RFC 2283, February 1998
3. S. Bhattacharyya et al.: A framework for Source-Specific IP Multicast Deployment. Internet Draft - <draft-bhattach-pim-ssm-00.txt>, July 2000
4. J-C. Bolot, T. Turletti, I. Wakeman: Scalable Feedback Control for Multicast Video Distribution in the Internet. In *ACM SIGCOMM'94*, pp. 58-67, London, UK, Sept. 1994
5. B. Cain, S. Deering, B. Fenner, I. Kouvelas, A. Thyagarajan: Internet Group Management Protocol, Version 3. Internet Draft - <draft-ietf-idmr-igmp-v3-07.txt>, March 2001
6. A. Campbell, G. Coulson, D. Hutchinson: A Quality of Service Architecture. *ACM Computer Communications Review*, April 1994
7. B. Fenner, M. Handley, H. Holbrook, I. Kouvelas: Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised). Internet Draft - <draft-ietf-pim-sm-v2-new-02.txt>, March 2001
8. GCAP Project IST-1999-10504: Global Communication Architecture and Protocols for new QoS services over IPv6 networks. Website: <http://www.laas.fr/GCAP>
9. B. Haberman, D. Thaler: Unicast-Prefix-based IPv6 Multicast Addresses. Internet Draft - <draft-ietf-ipngwg-uni-based-mcast-01.txt>, January 2001
10. H. Holbrook and B. Cain. Source Specific Multicast for IP. Internet Draft - <draft-holbrook-ssm-arch-02.txt>, March 2001
11. L. Mathy, G. Leduc, O. Bonaventure, A. Danthine: A Group Communication Framework. *Broadband Islands '94 Connecting with the End-User*, W. Bauerfeld, O. Spaniol, F. Williams, eds., Elsevier North-Holland, 1994, pp. 167-178
12. A. Meissner, L. Wolf, R. Steinmetz: A novel Group Integrity Concept for Multimedia Multicasting. *Proc. 8th International Workshop on Interactive Distributed Multimedia Systems (IDMS 2001)*, Lancaster, UK, September 2001
13. D. Meyer (ed.): Multicast Source Discovery Protocol (MSDP). Internet Draft - <draft-ietf-msdp-spec-08.txt>, April 2001
14. J. Nonnenmacher, E. Biersack, D. Towsley: Parity-based loss Recovery for Reliable Multicast Transmission. *IEEE/ACM Transaction on Networking*, Vol. 6, No.4, pp 349-361, August 1998
15. P. Owezarski, M. Diaz, C. Chassot: A time efficient architecture for multimedia applications. *IEEE Journal on Selected Areas in Communications*, Vol. 16, No. 3, pp. 383-396, April 1998
16. P. Owezarski, M. Diaz: New architecture for enforcing multimedia synchronization in videoconferencing applications. *Telecommunication System*, Vol. 11, No. 1-2, pp. 161-185, 1999
17. R. Vida, L. Costa, R. Zara, S. Fdida, S. Deering, B. Fenner, I. Kouvelas, B. Haberman: Multicast Listener Discovery Version 2 (MLDv2) for IPv6. Internet Draft - <draft-vida-mld-v2-00.txt>, February 2001
18. R. Vida, L. Costa, S. Fdida, R. Canonico, L. Mathy, D. Hutchison: Specification of a Multicast Monomedia Protocol. GCAP Project Deliverable D2.1.1, January 2001, available at <http://www.laas.fr/GCAP/public/publications.html>