Doc.#: AVC-797 (DRAFT!)

Title: Transmission of T.120 information in a LAN environment

Date: 09 May 1995

Purpose:Discussion and Proposal

Source: TU Berlin

Contact:Joerg Ott
phone +49 30 314-73389
fax +49 30 314-25156
jo@cs.tu-berlin.de

1. Introduction

The H.32Z series recommendations are concerned with extending videotelephony and teleconferencing services to include LANs. To accomplish this task, audio, video, data, and control information must be transferred between any number of terminals located on the LAN and the WAN(*). Due to the characteristics of LANs this information transfer will be done in some other fashion than is described in H.221 or H.22P

(*) In this contribution the term WAN only refers to connection-oriented (point-to-point) networks such as ISDN or PSTN. LANs include local networks built from individual LAN segments by bridges and/or routers.

While transmission of audio and video information across LANs in a packetized format is reasonably well understood today -- in point-to-point as well as in multicast environments --, the way of transmitting control and data information for tightly controlled conference settings has been ultimately solved only for point-to-point operation.

The present contribution aims at describing ways of distributing T.120 data information in a mixed multicast and point-to-point environment. It proposes a way to make use of LAN (and WAN?) multicast facilities where possible for more efficient information distribution, As a side effect this will reduce the network load and thus increase the probability of timely delivery of real-time information.

2. Interconnection Models

Figure 1 shows a complex scenario of how terminals may be interconnected through WANs and LANs, either in a point-to-point call or in a multipoint conference.

```
                        ________                        __________
     +-+      +-+      /        \   +-+      +-+      /          \   +-+
     | |      | |------  WAN #1  ---| |      | |-----  WAN #2    ---| |
     +-+      +-+      \________/   +-+      +-+      \__________/   +-+
      |        |         /           |        |          |  \
    ==+===+====+===     +-+        ===+====+===+===     +-+   +-+
          |             | |                |            | |   | |
  LAN #1 +-+            +-+        LAN #2 +-+           +-+   +-+
         | |                              | |
         +-+                              +-+
```

Figure 1: Interconnection Models for LANs and WANs

This figure could hold the following possible scenarios:

a) WAN-only terminals (only of interest for interoperability purposes; this has already been solved in T.120 in conjunction with H.320/H.324)
b) LAN terminal(s) connected to WAN terminal(s) through a gateway
c) LAN terminals within a single LAN only
d) LAN terminal(s) on several LANs interconnected by a WAN
e) WAN terminal(s) connected to WAN terminal(s); different WANs are used; the different WANs are interconnected through a LAN

The design of any transport for T.120 data information should consider the existence of all the above scenarios. This means that any extension of the T.123 protocol stacks has to be able to interoperate with all other T.120 terminals that do not implement this extension. As a corollary, the service offered by the T.122/T.125 Multipoint Communication Service must not be affected.

3. Protocol Stack Aspects

The current T.123 specification defines T.120 protocol stacks for various connection-oriented WANs. These protocol stacks all offer an X.224 service interface on which the T.122 service relies. The QoS of the point-to-point transport offered is

- flow controlled,
- error-free,
- sequence-preserving, and

delivery of packets that do not exceed a previously agreed size.

3.1 Current LAN Protocol Stacks

In Appendix II of T.123 three possible protocol stacks for LANs are proposed, each of them offering a QoS and service interface for point-to-point connection similar to the above. However, neither of these proposals exploits the multicasting facilities available on LANs, so all of them incur significant additional load on the LAN if the data information is to be transmitted to multiple terminals (or MCUs) in the LAN.

3.2 Multicast Service offered in LANs

In recommendation X.6 the ITU-T has defined the terminology and various service types for multicasting. Following this document, the suitable multicast facilities that can be used directly (i.e. without any additional protocols) from the service offered by a LAN can be characterized as follows:

- n-way communication
- non-real-time
- no explicit group creation and control (no especially authorized member)
- each group is identified by a single network address
- there is no membership control, i.e. such a group is always "open"
- the notion of a multicast call as specified in X.6 does not exist; a "multicast call" is implicitly established when the first host joins a group and is implicitly terminated when the last host leaves it; hosts may join and leave the call at will
- changes to the membership are "receiver-oriented", i.e. only JOIN and LEAVE are supported; there is no way to INVITE or EXCLUDE members from a call

- the following attributes apply to data transfer
  - data integrity is not guaranteed (packet loss and corruption may occur)
  - data delivery is not synchronized with respect to a certain time window
  - data is delivered in any order (note that X.6 specifies at least per source ordering as required)
  - throughput is not specified; the sender determine the speed of the transmission
  - there is no flow control; receivers that cannot keep up with the pace of the transmission miss packets

Obviously, the service offered by a LAN for multicasting cannot be used directly below MCS. What is needed to fill the gap between the services and the QoS offered by the LAN, and the needs of the MCS is some layer XYZ between MCS and the LAN services (figure 2).

```
+---------------+
|  T.122/T.125  |
+---------------+
|      XYZ      |
+---------------+
|      IP       |
+---------------+
| e.g. Ethernet |
+---------------+
```

Figure 2: Protocol Stack for T.122 in a Multicast LAN

3.3 Outline of the XYZ Layer

The XYZ layer must provide at least the following functionality

a. It must make multicasting as reliable as are point-to-point transport connections otherwise used by T.122. This means that the QoS must be improved to reliable, ordered, and flow-controlled multicast service to allow exchange of MCSPDUs without any additional measures to be taken within the MCS.

b. The receiver-oriented multicast service needs to be extended to support INVITE and EXCLUDE as well, to be able to properly support all kinds of connection setup.

c. Several membership services need to be included to allow detection of new members, lost "connections" to certain members, and the like.

d. A negotiation service must be included that allows to determine the mode of communication between two systems (WAN, LAN-multicast, LAN-point-to-point).

The design of the XYZ layer must take into account a very important prerequisite: wherever in the topology depicted in figure 1 one or more LANs are located, neither the MCS service nor the protocol behavior of an MCS entity to the outside of the LAN must be changed in any way. The hierarchy of the MCS connections remains but LAN islands may be found at various levels in branches of the MCS tree. Non-LAN connections must not experience any difference in the protocol. This implies that enhanced or modified MCSPDUs or specific XYZ-PDUs may only be exchanged within a LAN, that state information is local to each LAN (and independent from others).

4. Proposed Solution for the XYZ Layer

We propose to divide the functionality of the XYZ layer into two parts:

* a (set of) generic multicast transport protocols that offers reliable, at least single source ordered, and flow-controllable service; and

* an adaptation layer that provides the additional T.120/H.23Z specific functionality. Where necessary this layer may also make use of reliable point-to-point protocols e.g. for peer negotiations and "connection" setup.

This approach could benefit from the availability of a general-purpose multicast transport protocol -- as is referred to in the Annex as well --, that might become standardized(*) in the future and allow our working group to focus on the development of the adaptation protocol and on the integration into the H.32Z protocol suite.

(*) Note, however, that while this is clearly desirable and several groups are working on reliable multicast transport protocols, currently there seems to be no standardization effort on this subject, neither in the ITU nor in the IETF.

A specification how such a XYZ-Layer can be implemented is given in the Annex to the present contribution. The attached paper was presented at the International Conference on Communication Systems (ICCS'94) in November 1994.

5. Work Items

The following items are at least to be addressed in this working group to complete the specification for how to transmit T.120 data information across LANs:

- Are T.120 "connections" integrated in the H.22Z setup and control mechanism and, if so, how?
- How is bandwidth allocated to T.120 connections?
- Security against unauthorized participation (can mainly be achieved by means of data encryption on LANs; is required for both point-to-point and multicast connections. because eavesdropping on LANs is always possible and quite simple)

Besides this there is the open issue when a reliable multicast transport becomes standardized.

6. Conclusion

The present contribution has outlined a way how T.120 data information could be transmitted in LANs by exploiting their multicast capabilities.

The item to be filled in with this proposal is that there is no multicast transport protocol standardized at the moment. However, the approach presented shows that multicasting MCSPDUs is basically possible without any changes to the MCS service. Furthermore, especially in LANs with no QoS guarantees, saving traffic on the network will be beneficial to the audio and video information streams.

Therefore, it is proposed to define a service mapping based on an abstract multicast service that offers sufficiently reliable, ordered, and flow-controlled QoS for information transmission within (arbitrary) large groups. Since achieving these properties has been the goal of many research activities, a standard to match these requirements is likely to be established in the near future. Until then the choice of which reliable multicast protocol (if any) to support remains up to the manufacturers.

To have a fully standardized solution for the short term a fan-out of reliable point-to-point connections (as suggested by T.123/Annex II) based e.g. on the TCP/IP protocol suite should be defined. This should include at least a defined way to negotiate the usage of a multicast protocol (be it proprietary or standardized) to allow future extensions to be integrated in a straightforward manner.

7. References

[1] AVC-757. Draft of H.22Z. Media Stream Synchronization and Time Base Recovery on Non-Guaranteed Quality of Service LANs. 28 April 1995.

[2] ITU-T Recommendation H.221. Frame Structure for a 64 to 1920 kbit/s Channel in Audiovisual Teleservices. 1993.

[3] Draft ITU-T Recommendation H.22P. Multiplexing Protocol for Low Bit-Rate Multimedia Communication. 1995.

[4] ITU-T Recommendation T.122. Multipoint Communication Service for Audiographics and Audiovisual Conferencing -- Service Definition. 1994.

[5] ITU-T Recommendation T.123. Protocol Stacks for Audiographic and Audiovisual Teleconference Applications. 1994.

[6] ITU-T Recommendation T.125. Multipoint Communication Service -- Protocol Specification. 1994.

[7] ITU-T Recommendation X.6. Public Data Networks: Services and Facilities. Multicast Service Definition. 1993.

Annex -- [PostScript document formatted for 297x210 mm paper]

MULTICASTING THE ITU MCS: INTEGRATING POINT-TO-POINT AND MULTICAST TRANSPORT

JÖRG OTT
Technische Universität Berlin
FB 13, Sekr. FR 6-3, Franklinstr. 28/29, D-10587 Berlin, Germany; jo@cs.tu-berlin.de

CARSTEN BORMANN
Universität Bremen
FB 3 AG Rechnernetze, Postfach 330440, D-28334 Bremen, Germany; cabo@informatik.uni-bremen.de

ABSTRACT Many designs for multicast transport protocols presume a homogeneous view of the network environment they are operating in, the latter hiding the various networks' features from the transport layer. However, knowing its underlying network characteristics enables a multicast transport protocol to adjust itself to particular networks. This can be important for providing a scalable, reliable multicast transport. A multicast group often is faced with different topologies (point-to-point links vs. broadcast capable networks) and different network service qualities (e.g. packet loss probability, throughput, and delay): For some parts of the group, network connectivity allows multicasting and the usage of negative acknowledgment schemes, for others, point-to-point connections with positive ACKs are required. Trying to unite both service types within a single type of transport is inefficient and often not possible at all. We present a hybrid approach to a multicast transport that extends a standardized multicast service based on point-to-point connections to be efficiently applicable in multicast capable networks as well.

I. INTRODUCTION

Recently, interest in multiparty teleconferences and group cooperation has tremendously increased, and, after laboratory experiments, teleconferences now begin to be widely employed in daily business [6].

Various teleconferencing systems have been designed covering a wide range of collaboration scenarios from tightly-coupled conferences of a handful of participants to loosely-coupled broadcasts of meetings with the remote listeners having the chance to intervene.

All types of teleconferencing systems have in common that they require a transport protocol for the conference-wide exchange of information. In addition to real-time data, control information and other non real-time data must be transferred as well: This is our focus of interest. For group communication such a transport should simplify the application design by being easy to use, reliable (if required), and commonly accepted — just as TCP is for point-to-point connections. A simple fan-out of TCP connections obviously does not scale to large groups: packets need to be transmitted repeatedly which is unnecessary in networks capable of multicasting. What is needed is a multipoint transport that employs underlying multicasting facilities if available but can also work without them otherwise.

In addition to numerous research projects, standardization bodies such as the *International Telecommunication Union (ITU)* and the *Internet Engineering Task Force (IETF)* are concerned with multiparty conferencing and the corresponding multicast transport. The teleconferencing scenario of the ITU assumes Multimedia Terminals (MMTs) interconnected via one or more Multipoint Control Units (MCUs) by means of public switched networks such as ISDN. The IETF proposes a multicast datagram platform that hides the network structure [8]. In the IETF, no choice for a multipoint transport has been made so far. The ITU has defined a multipoint protocol tailored to their specific needs: the Multipoint Communication Service, MCS [10, 12], based on point-to-point transport connections.

Our aim is a communication infrastructure for multiparty conferences with both standard conformance and flexibility. A typical teleconferencing scenario we have in mind consists of several sites — interconnected via WANs — each of which may involve several participants via LANs. MCS offers a clearly defined service interface meeting many of our requirements. However, as the MCS is designed for point-to-point communication environments it is inefficient when used in multicast capable networks such as LANs.

We describe an extension to the MCS protocol that makes it applicable on top of a multicast network while retaining standard conformance across point-to-point connections. A prerequisite for our extension is the availability of a reliable multicast transport.

In the following section, we briefly introduce MCS; in section III, our choice for a reliable multicast transport: MTP-2. In section IV, we present the overall integration model, followed by a detailed description of the necessary protocol extensions in sections V and VI. In Section VII we give an example for a communication process involving our extensions, and in section VIII we discuss the costs and benefits introduced by our extensions.

II. THE MULTIPOINT COMMUNICATION SERVICE

The Multipoint Communication Service (MCS) supports flexible communication facilities within a teleconference — the *domain* — that is built by interconnecting a set of MCS entities — the *agents* — in a tree structure. Its root — the *Top MCS (agent)* — coordinates concurrent operations. MCS provides *channels* for addressing within the conference. A channel uniquely identifies an application entity (for unicasting) or group (for multicasting). Information delivery is error-free, flow-controlled, and sequential (globally or with respect to a single sender) for each channel. A token mechanism is provided to coordinate applications.

From the underlying (point-to-point) transport, MCS expects a connection-oriented, error-free, reliable, and flow-controlled data transfer service according to the profiles defined in T.123 [11]. MCS uses up to four transport connections per peer agent pair to transmit information of four different priorities.

III. CHOOSING A RELIABLE MULTICAST TRANSPORT: MTP AND MTP-2

A multicast transport for MCS must provide similar guarantees, and it must be scalable to work efficiently also in larger groups. We have chosen MTP [1] as a starting point for a multicast transport protocol because it approximately matches these requirements.

Using negative acknowledgments, MTP avoids *ack implosions* [13]. It provides rate controlled, totally ordered, atomic exchange of information within a group (the *web*). MTP uses a single *master* to achieve rate control, ordering and atomicity as well as to increase reliability. The other members of a web are *producers* and *consumers*. Producers are potential originators of information, consumers are only recipients. Information transfer is done by means of messages each of which consists of a sequence of one or more packets. The master has to be contacted prior to each message transmission; it grants a token along with a sequence number for each message and acknowledges its receipt. In networking environments with a low error rate, MTP has proven to work reliably even under heavy load [2]. We have improved MTP to become more general and to be able to serve as a suitable platform for MCS. Among other changes, MTP-2 [3] introduces unicasting and the ability to change the master of the web.

IV. EXTENSIONS TO THE MCS PROTOCOL

To ensure interoperability with non-extended MCS implementations we do not change the service or protocol semantics of the MCS specifications. In particular, this applies to the exchange of (non-standard) control PDUs: during connection establishment this is done out-of-band with respect to the regular flow of MCS PDUs, and during operation of a domain the extended / additional PDUs are kept local within each web. An *MCS to MTP Adaptation Protocol* (MMAP[1]) is provided on top of MTP-2 to implement the functionality that is required for this particular application but is not sufficiently generic to become part of MTP-2 itself.

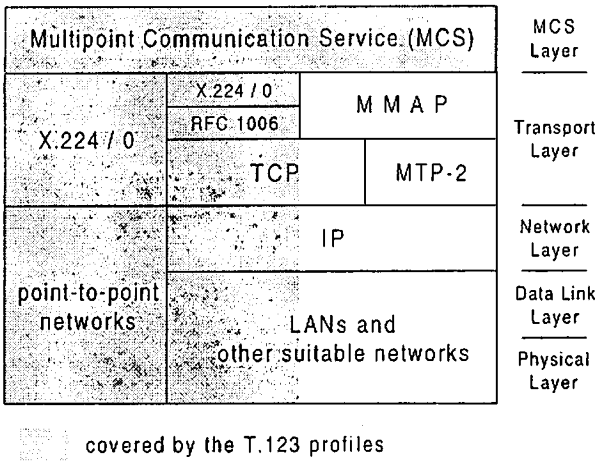

Figure 1: Protocol stacks for the extended MCS design

As depicted in Figure 1, MMAP is based on top of both TCP and MTP-2. TCP is ubiquitous in IP based (LAN) environments and, therefore, is used in addition to MTP-2[2] for the setup procedure (see section V). Furthermore, the TCP connection is used for the initial data transfer to avoid setup delays. Such delays are otherwise introduced by MTP-2 having to wait for an agent to become master of a web and the peer agent to join this particular web. After a web has been successfully established, the TCP connection is abandoned and the entire communication is carried out using MTP-2. If the web could not be established (if there is insufficient multicast connectivity), TCP is used as transport instead (with RFC1006 [14] and X.224 class 0 [7] on top as shown in Figure 1).

In our model, MTP-2 is only used for communication in network areas (such as LANs) that support multicast and have sufficiently controlled delay and error rates to use negative acknowledgment protocols; otherwise, or if interoperation with a non-extended MCS implementation is needed, a point-to-point transport protocol is used.

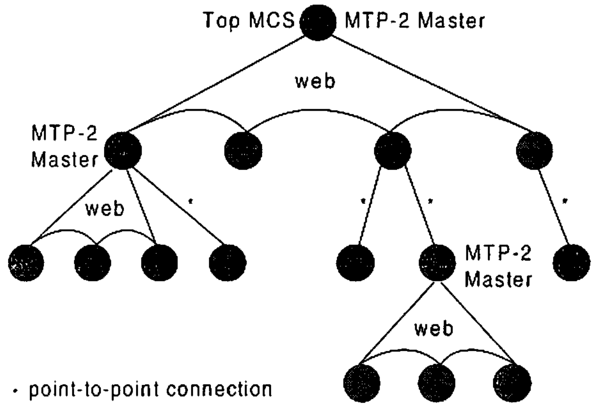

Figure 2: Using local multicast webs in the MCS tree structure

MCS agents connect to each other, one by one, to set up the tree structure of the domain. To decide whether the "peer" MCS is reachable via multicast or not, we introduce the notion of a *multicast horizon*. Peer agents that are located "within this horizon" are included in local MTP-2 webs, others are connected via a conventional transport protocol. Each multicast web only spans a single level of a branch of the MCS tree. The MTP-2 master role is always performed by the superior MCS agent (Figure 2).

We split the following more detailed discussion of MMAP into two areas: setting up a multicast transport instead of point-to-point connections (section V), and mapping the MCS protocol onto the multicast transport during the operational phase (section VI).

V. CONNECTION SETUP

An MTP-2 capable MCS agent (the calling agent, *C*) wishing to connect to another MCS agent (the accepting agent, *A*) does not know in advance whether its peer is MTP-2 capable as well. Neither does *C* know whether the multicast connectivity between the two agents is sufficient for a particular MCS domain (with its throughput requirements) or for running a NAK based transport protocol at all. Consequently, *C* must be able to contact *A* and determine *A*'s capability to run MTP-2 and find out about the multicast connectivity. *C* and *A* must agree on a common set of QoS parameters that is suitable for the MCS domain. This must be performed out-of-band to the regular MCS information flow. Setup failures of the multicast web must not have any impact on the MCS service if point-to-point communication is possible. Webs must be created according to the MCS hierarchy, and newcomers must be added to the correct existing web (with respect to their position in the hierarchy relative to their peer), or a new web must be created. In Figure 3, a particular case of this process is illustrated.

[1] With apologies to the inventors of the Berkeley UNIX system call of the same name, *mmap(2)*.

[2] Another protocol such as TCP is required because the unicasting facility of MTP-2 is unavailable unless a web is established and both parties are members of the web. The only alternative would be re-implementing some protocol similar to TCP independent from a web in MTP-2.

A. Connection Setup Process

An MCS agent binds to the MCS TSAPs for the various networks and network addresses through which it is reachable. An MTP-2 capable agent also binds to a (well-known) TCP port for MMAP. By simply connecting to this port and exchanging an initial pair of PDUs the initiator C of the connection may determine whether its peer A is capable of communicating via MTP-2. C generates a multicast address (see below) and tries to create a new MTP-2 web, itself being the master. Then, C sends a SETUP PDU through a newly established TCP connection and includes the QoS parameters its MTP-2 web and the MCS agent are able to handle[3]. A responds with an ACCEPT, possibly modifying the QoS parameters (again, according to its agent's capabilities or the domain's requirements), or REFUSEs the connection setup. These pure MMAP PDUs are comparable to the connection setup handshake PDUs of other protocols.

After initial contact, C initiates the MCS connection setup handshake (as is usually done for point-to-point connections). The MCS PDU is encapsulated in an MMAP PDU that also carries additional MMAP information. Besides other purposes, this part of the MCS protocol allows C and A to find out who is to be the superior and who is to be the subordinate node in the MCS tree. Based on this outcome, C and A decide whether they need the new MTP-2 web for communication (i.e., with only two members for the beginning) or whether they may use an existing web. If A is the superior node[4] and there is already an MTP-2 web active for A's subordinates within the same multicast horizon, then C is informed about the multicast address (INVITE) of the existing web. C discards its new web and joins the existing one. If no suitable web exists the newly created one is used. C awaits the confirmation to be master of the new web and then INVITEs A specifying the multicast address for the new web. If A joins the web this is indicated to C which then multicasts a confirmation to all web members (JOIN-CONFIRM). When the confirmation is received by the new member agent, it initiates the release of the TCP connection (SETUP-COMPLETE) as the join and confirmation messages indicate multicast connectivity in either direction. If A is to be the superior node then a master change is initiated by C. INVITE, JOIN-CONFIRM, and SETUP-COMPLETE, again, are pure MMAP PDUs and do not carry any MCS related information.

Note that it takes some time from the reception of the MCS connection setup confirmation to the MMAP setup completion. Besides other issues, this duration depends on the selected MTP-2 parameters and the mechanism for acquiring a multicast address. During this time period the MCS connection should not be blocked; otherwise the higher layer connection setup process would be slowed down noticeably. Therefore, further MCS PDUs are transmitted — encapsulated in MMAP PDUs — using the TCP connection until the MTP-2 setup has completed. This may cause some PDUs being transmitted through MTP-2 as well as through TCP. To detect duplicates at the receiver, the MTP-2 sequence numbers are included in the MMAP headers as well (and thus also sent via TCP).

B. Assignment of Multicast Addresses for MTP-2 webs

An important issue for the efficient operation of MTP-2 is the exclusive availability of a multicast address per web within the multicast horizon[5] — a problem common to all applications based on the current IP multicasting. The way of determining whether a multicast address is free to use directly impacts the creation time for a web ($t_2 - t_0$ in Figure 3). MTP-2 considers a multicast address to be available if after several requests to this multicast address no response has been received.

If a service for the allocation of multicast addresses is available (e.g. as suggested by Braudes and Zabele [5]), an MMAP entity can use this service to obtain a free multicast address. This would eliminate the delay introduced by the MTP-2 timeout mechanism. Address allocation may also be implemented with a local scope if non-interference with non-local multicast traffic can be guaranteed (e.g. by means of multicast firewalls).

VI. OPERATIONAL PHASE: MAPPING THE MCS PROTOCOL ONTO MTP-2

A. Data Transfer

Using a multicast transport may violate assumptions that are valid only for point-to-point connections. Multicasting MCS PDUs through a web requires their flow direction and their impact on the traversed agents at the MCS layer to be considered for the mapping. Taking into account these aspects, we have identified several groups of MCS PDUs and apply multicasting as follows:

1) *Unordered transmission.* MCS implements the unordered data distribution by means of distribution lists at each non-leaf node. A data PDU destined to a certain MCS channel is forwarded upwards until it reaches the Top MCS. In parallel, each node also forwards it downwards to all those peer agents that have members of the respective channel in their subtree. Data PDUs are not forwarded to peers they were received from.

With MTP-2, unordered data transmission is done by simply multicasting the encapsulated MCS PDUs into the web. As global ordering is not guaranteed, this extension does not violate standard conformance. Different MCS priorities are mapped onto different MTP-2 streams.

Since an MTP-2 web only uses a single multicast address, all data PDUs sent through a web are received by all its members — even if this member is not in the distribution list. To allow the data PDUs not destined for a certain agent to be detected and discarded efficiently, we map the MCS distribution list onto a subgroup identifier that is carried in the MMAP header.

2) *Globally ordered transmission* of information is ensured in MCS by having the Top-MCS agent sequence the flow of data PDUs. Consequently, in MCS all such data packets need to be transmitted uptree to the Top MCS that initiates the distribution back downtree — again, this is done according to the distribution lists.

The extended MCS cannot deviate from this structure as this would violate the global ordering property[6]. Therefore, we retain the two phases but employ multicasting: on the way upwards all members of a web receive a copy of the encapsulated MCS DATA PDU transmitted through the web and store its globally unique id (located in the MMAP header) and contents. In the following downtree distribution phase, multicasting is used for distribution of the MCS PDUs to subtrees they

[3] If the MCS agent is already member of a domain, the requested QoS parameters are dictated by the MCS domain parameters.

[4] The procedure is basically the same if C is the superior node.

[5] MTP-2 operation is not disturbed by other applications using the same multicast address because foreign packets are unlikely to match the long MTP-2 web identifiers and would thus be discarded by MTP-2. The drawback is that more packets would have to be processed by MTP-2.

[6] Obviously, if the MCS tree consists of only a single hierarchy level and is made up of a single MTP-2 web, the global ordering property of MTP-2 may be applied. We do not make use of this special case.

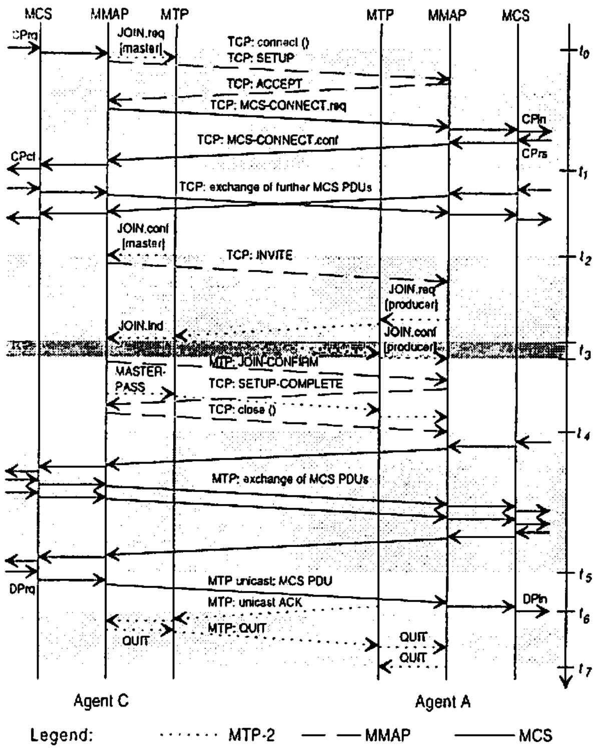

Figure 3: Sample Communication Scenario

did not originate from — just as in the unordered transmission. In webs that have already been traversed by this particular data PDU on its way upwards, only the packet id is passed through the web — an approach that is useful especially for large data portions.

3) *MCS internal management information*, such as (changes in) channel membership, token status, etc. is originated from the Top MCS and forwarded downwards through all the transport connections. With MTP-2, these PDUs are simply encapsulated in MMAP PDUs and distributed by multicast through each web on their way downwards.

4) *Unicast MCS PDUs* — such as requests and unicast data PDUs — are routed through the MCS tree only to the particular agent the information is destined to. In the extended MCS, these PDUs are encapsulated in MMAP PDUs and transmitted using the MTP-2 unicasting mechanism.

Our extension to MCS does not require all the functionality offered by MTP-2: Atomicity of information delivery is not required; neither is the global ordering property. Omitting these functions causes a reduction of latency as a recipient neither has to wait for the master to confirm a message nor for a previous message to be completed.

B. Disconnect handling

The MCS service expects the underlying transport service to notify it when a peer agent disconnects. A web combining any number of peer agents must provide disconnect notifications as well. To accomplish this task, MMAP maintains a membership list at the MTP-2 master.

Finding out about systems leaving a web is only partially supported by MTP-2 itself. Before an agent leaves, it sends an MTP-2 QUIT packet to the master who then knows about the change. However, if a member crashes, is partitioned away, or the QUIT packet is lost, the master is not informed. Hence, MMAP incorporates ALIVE messages that are exchanged regularly to enable the master to detect silent failures of MCS agents.

If the superior MCS node (i.e. the MTP-2 master) leaves, all "transport connections" have to be destroyed. The other web members are informed by an MMAP FORCE-QUIT PDU sent by the master. If a master fails, this message is generated by the new master after the master recovery procedure of MTP-2 has completed[7]. If only a single member (the master) remains in a web it also leaves.

VII. A SAMPLE COMMUNICATION SCENARIO

Figure 3 depicts a scenario for running the MMAP protocol to successfully set up a new MCS connection for use with MTP-2. In this example, no existing web is used. In the new web, a master change occurs. Furthermore, two MCS PDUs traverse the new TCP connection while the MTP-2 connection setup is still in progress. After successful setup of the web four further MCS PDUs are exchanged. Finally, C leaves the web again which causes A to leave and destroy the web.

At t_0 an MCS connect request (CPrq) is issued by the calling agent C, indicated (CPin) to and answered (CPrs) by its peer A. The setup completes from the point of view of the MCS with C receiving the confirmation (CPcf) at t_1. In this scenario two MCS PDUs immediately follow the setup. These are exchanged via TCP as the web creation does not complete before t_2 and no data exchange via MTP-2 is possible before both agents are joined to the web (t_3, seen from the viewpoints of C and A, respectively). The communication is performed solely through MTP-2 from t_4 onwards. At t_5 C issues an MCS disconnect request (DCrq) which is sent via unicast and indicated to A (DCin) at t_6 — at this time the connection release is complete from the MCS point of view. After MTP-2 has confirmed that the message has been successfully delivered, C leaves the web, which — as no other members remain — is destroyed by the master A. At t_7 the web is gone and the multicast address may be reused.

VIII. COSTS AND BENEFITS OF MMAP

Benefits and overhead may be measured in processing time, number of packets, and transmission delays all of which are assessed for the connection setup and the operational phase, respectively.

A. Connection Setup Phase

In the connection setup phase, none of the benefits of multicasting becomes visible. On the contrary, additional overhead is required. Our goal for the setup phase is to minimize the overhead, in particular, the noticeable delay.

For setting up a transport, a TCP connection and an MTP-2 web have to be established. In terms of packet overhead, the TCP connection requires an initial handshake and some further MMAP control PDUs (the encapsulated MCS PDUs transmitted via TCP are sent anyway). MTP-2 requires some overhead for finding a free multicast address and for membership control. However, all these packets are transmitted only once per new member (and in some cases not even the web setup is needed).

MMAP introduces no additional delay compared to standardized MCS transport platforms. The potential web setup delay is unnoticeable as MMAP uses the TCP connection for information exchange prior to completion of the setup. The

[7] There is no recovery procedure defined in MCS in case of failure of the superior node.

single round trip for negotiating capabilities is present in OSI transport protocols and thus not specific to MMAP (with which we may even eliminate this delay in the future).

Processing overhead may be neglected for the setup messages as these are sent only once per member (as are setup messages of other transport protocols).

Finally, MTP-2 requires an unused multicast address that needs to be allocated either by trial-and-error (which is time and bandwidth consuming) or is requested from some kind of assignment service. In the latter case — in particular if the service is a local one — address allocation becomes quick and cheap.

While the protocol may already be "good enough" as presented, we are now considering several potential further optimizations, such as performing the initial capability exchanges along with the MCS connect PDUs, and delaying the creation of a multicast web until at least three agents can participate.

B. Operational Phase

In the operational phase, we distinguish between unicast and multicast transmissions. The latter can benefit from using MTP-2 because the number of packets to be transferred is reduced if — and this is the assumption within a multicast horizon — the number of lost packets may be neglected. With no packet loss, in a web with n members only *one* instead of n packets carrying identical information are sent. Furthermore, as opposed to TCP — an implementation of which should generate an acknowledgment for at least every second packet received [4] — in case of successful transmission of all packets no acknowledgments are needed at all. Note that this also applies to globally ordered MCS PDUs, except that two messages are multicast in a web that is traversed on the way up the tree.

A producer must ask the master for a token before it actually can transmit a message, thus adding two packets and one round-trip delay for each message. To cope with this drawback, MTP-2 allows piggybacking of token requests on previously sent messages, reducing the overhead to one packet. Furthermore, much of the information distribution in an MCS tree will be top down (at least for webs at lower hierarchy levels) so that master and producer are identical: in this case the token request overhead is reduced to some internal function call.

For unicasting, using MTP-2 does not provide benefits as it works with positive ACKs and is comparable to TCP.

As a pure overhead, MMAP introduces ALIVE messages to keep a consistent membership list at the master. The bandwidth consumption is kept below a maximum fraction of the bandwidth available for the entire web by deriving the transmission interval from the number of web members.

Apart from the token overhead discussed above, the remaining functions of MTP-2 that might cause further delay — atomicity and global ordering — are not used for MMAP.

Regardless of the transmission type employed, MMAP requires additional processing that, however, is minimized by having a very simple protocol with fixed sized headers. The processing overhead for running MTP-2 is comparable to the processing required for e.g. the TCP, RFC1006, and X.224/0 stack.

IX. CONCLUSION

MCS, originally intended for point-to-point links, has been enhanced to benefit from networks that support multicasting. This may be a starting point for standard conforming integration of WAN and LAN teleconferences, a typical scenario for collaboration. The approach to extending MCS works well and the design is a good compromise between network traffic and (additional) protocol processing.

Currently, our approach is primarily applicable to LANs, as the prerequisites for wide area multicasting — such as resource reservation [15] and sparse mode multicast routing [9] — are not yet widely deployed in the Internet. When these technologies are available, an MCS tree may be reduced to a single MTP-2 web allowing efficient communication as well as offering a standard conforming service for a reliable multicast transport.

X. ACKNOWLEDGMENTS

We would like to thank Hans-Christian Gehrcke, Torsten Kerschat, and Nils Seifert of TU Berlin for the prototype implementations of MTP and MTP-2 and their participation in the development of MTP-2 as well as Andreas Schmidt for the implementation of MCS. In various discussions, all of them have contributed to the concept of integrating MCS and MTP and the respective software designs.

REFERENCES

1. S. Armstrong, A. Freier, and K. Marzullo, "Multicast Transport Protocol," Internet RFC 1301, February 1992.
2. C. Bormann, J. Ott, and G. Hoffmann, "First Experience with Multicasting the X Protocol," in *Broadband Islands Third International Conference*, Elsevier, 1994.
3. C. Bormann, J. Ott, H.-C. Gehrcke, T. Kerschat, and N. Seifert, "MTP-2: Towards Achieving the S.E.R.O. Properties for Multicast Transport," to be presented at the 1994 International Conference on Computer Communications and Networks (ICCCN '94), 1994.
4. R. Braden, Ed., "Requirements for Internet Hosts — Communication Layers," Internet RFC 1122, October 1989.
5. R. Braudes and S. Zabele, "Requirements for Multicast Protocols," Internet RFC 1458, May 1993.
6. S. Casner and S. Deering, "First IETF Internet Audiocast," *ACM Computer Communications Review* 22(3), July 1992.
7. CCITT Recommendation X.224, "Transport Protocol Specification for Open Systems Interconnection for CCITT applications," in *Blue Book, Fascicle VIII.5*, CCITT, Geneva, 1988.
8. S. Deering, "Host Extensions for IP Multicasting," Internet RFC 1112, August 1989.
9. S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu, and L. Wei, "An Architecture for Wide-Area Multicast Routing," *Computer Communications Review* 24(4):126–135, ACM, August 1994.
10. ITU-T Draft Recommendation T.122, "Multipoint Communication Service for Audiographic and Audiovisual Conferencing, Service Definition," ITU-T, 1993.
11. ITU-T Draft Recommendation T.123, "Audiovisual Protocol Stack," ITU-T, 1993.
12. ITU-T Draft Recommendation T.125, "Multipoint Communication Service, Protocol Specification," ITU-T, 1993.
13. M. G. W. Jones, S.-A. Sørensen, and S. R. Wilbur, "Protocol design for large group multicasting: the message distribution protocol," *Computer Communications* 14(5):287–297, June 1991.
14. M. T. Rose and D. E. Cass, "ISO Transport Service on top of the TCP, Version 3," Internet RFC 1006, May 1987.
15. L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: A New Resource ReSerVation Protocol," *IEEE Network* 7(5):8–18, September 1993.