March 1993

The Integrated Media Approach to Networked Multimedia Systems

Klara Nahrstedt
University of Pennsylvania

Jonathan M. Smith
University of Pennsylvania, jms@cis.upenn.edu

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-93-35.

Abstract

Applications which require real-time multimedia services[13] face a number of difficult problems in the transmission of multimedia information. Among the most difficult problems are the heterogeneity of end nodes and the heterogeneity of media *Quality of Service* (QoS) requirements. End nodes typically consist of a computer and number of sensory input and output devices, such as displays, microphones, and cameras. QoS requirements[18] include degrees of reliability, jitter, and delay.

We propose an *integrated* approach to address these problems. Multimedia input data comprise a sensory environment which an application will make available; these data are packaged together into an Integrated Multimedia Message (IMM). From a received IMM, output data are selectively reproduced to create another sensory environment. We propose an IMM format and protocol behaviors for generation, presentation, and synchronization of these messages.

While IMM's are aesthetically pleasing, well-suited to proposed high- speed networks, and ease intramessage synchronization, they are potentially plagued by the need to deliver QoS which meets the worst-case requirements of all of their components[6]. We believe that this problem can be addressed, and are testing that belief experimentally with the U. Penn Experimental Multimedia Conferencing System, which will be embedded in the AURORA Gigabit Testbed.

The Integrated Media Approach To Networked Multimedia Systems

MS-CIS-93-35
DISTRIBUTED SYSTEMS LAB 32

Klara Nahrstedt
Jonathan M. Smith

University of Pennsylvania
School of Engineering and Applied Science
Computer and Information Science Department
Philadelphia, PA 19104-6389

March 1993

The Integrated Media Approach to Networked Multimedia Systems

Klara Nahrstedt
Jonathan M. Smith*

Distributed Systems Laboratory
University of Pennsylvania, Philadelphia, PA 19104-6389

February 24, 1992

Abstract

Applications which require real-time multimedia services[13] face a number of difficult problems in the transmission of multimedia information. Among the most difficult problems are the heterogeneity of end nodes and the heterogeneity of media *Quality of Service* (QOS) requirements. End nodes typically consist of a computer and a number of sensory input and output devices, such as displays, microphones, and cameras. QOS requirements[18] include degrees of reliability, jitter, and delay.

We propose an *integrated* approach to address these problems. Multimedia input data comprise a sensory environment which an application will make available; these data are packaged together into an Integrated Multimedia Message (IMM). From a received IMM, output data are selectively reproduced to create another sensory environment. We propose an IMM format and protocol behaviors for generation, presentation and synchronization of these messages.

While IMMs are aesthetically pleasing, well-suited to proposed high-speed networks, and ease intramessage synchronization, they are potentially plagued by the need to deliver QOS which meets the worst-case requirements of all of their components[6]. We believe that this problem can be addressed, and are testing that belief experimentally with the U. Penn Experimental Multimedia Conferencing System, which will be embedded in the AURORA Gigabit Testbed.

1 Introduction

Current developments in the area of high-speed computer communications networks[20], as well as advances in related technologies will allow multiple high-bandwidth media applications on a single network fabric. The most important related technologies are:

1. *Video Technology*, such as High-Definition Television (HDTV) and displays with millions of pixels and many bit planes of color,

2. *Media-Dependent Compression Techniques*, such as JPEG and MPEG[5]; rather than viewing compression as removing the need for bandwidth, we look at compression as enabling the use of more cameras within the constrains of the bandwidth available,

3. *Processor Technology*, including fast RISC CPUs, larger DRAMs, larger address spaces and wider internal data paths [8],

*Research support for this work came from Bellcore (through Project DAWN), and from the Corporation for National Research Initiatives (CNRI), which is funded by the National Science Foundation and the Defense Advanced Research Projects Agency. Prof. Smith was also partially supported by an IBM Faculty Development Award.

4. *Window Systems*, which allow display usage to be controlled and customized; among the many uses of such systems is the multiplexing of a display across several concurrent applications,

5. *User Interfaces*, including mice, joysticks, pointers, microphones, cameras and software which manage these devices,

In this paper we focus on a design for *Networked Multimedia Systems* which addresses some current research problems, and which is sufficiently flexible to cope with future developments.

Flexibility of any design is particularly important, and we see a number of important application areas for non-traditional "media" (as compared with traditional media such as Audio, Video, Images and Text[13]) which should be accommodated by a truly "multimedia" system:

- tactile data, which is analogous to the human sense of "touch", has proven to be tremendously useful in robotics applications (Peter Allen thesis) where material characteristics can be used to determine whether or not a grasped object will be damaged, its weight, etc. Telerobotics applications allow humans access to hazardous environments, and a general multimedia system should bring all possible data to a remote human operator.

- machine analogues to taste and smell (e.g., with chemical sensors) which in fact can be reproduced remotely; "scent disks" are commercially available!

- non-human "senses" which might usefully be translated to human-usable forms. For example, radioactive emission levels as detected on a Geiger counter can be displayed in color map analogues or as temperature increases.

It is important to build in such flexibility at the design stage, as the design of Internet Protocol (IP) demonstrates (whatever faults one sees with IP[19], it runs across an incredible number of networking substrates).

Traditional Multimedia systems have been constructed by combining networks specialized for particular media. For example, teleconferencing systems for workstations are typically composed of CATV for video, telephony or an ISDN B-channel for audio, B-ISDN[26] for audio and video, and a LAN or ISDN D-channel for data and control information. While the details may vary, e.g., video compression may be used so that the LAN has adequate bandwidth to provide video services, the basic structure of distinct message streams is maintained from end-application to end-application, across distinct media. There are currently compelling cost/performance advantages in such an approach, due mainly to engineering experience, performance advantages due to specialization, and availability. For example, coaxial cable can easily carry fifty channels of uncompressed NTSC format video.

However, with the advent of very high-speed networks and media-rich environments, these traditional approaches fall short in a number of respects. In particular, they:

- require distinct cabling

- require separate stream handling (e.g. different transport protocols)

- are not completely digital

- have difficulty integrating new media

- force separate synchronization of each medium for presentation

- assume a lower-bound of input and output capability at each end-node

Our approach is based on the integration of the different media on the sender side in a novel data format (to be discussed in detail in Section 3.1). Data maintain this integrated form end to end through the network until they are received. We propose a design of an *Integrated Multimedia Message* (IMM) format and the protocol behaviors for generation, synchronization and presentation of these messages. It would be premature to claim that the integrated media approach to networked multimedia is measurably better or worse than other approaches. Rather, we wish to present a careful analysis of issues and a different approach to dealing with these issues. This approach, which we call the integrated approach, has strong advantages, especially for real-time networked multimedia systems, when compared to other approaches using B-ISDN or similar service provision.

1.1 Related Work

We will concentrate on related work in real-time networked multimedia systems (NMS), although there is also work in non-real-time NMS. Non-real-time systems are employed in storage and retrieval of multimedia documents, distributed processing of multimedia documents etc. [28], [21], [4], [30].

In real-time NMS, however, the integration for all media is at the user interface. Systems differ in their organization of data formats and data transmission approaches.

The first approach is to *transmit the different media in separate streams* and have for each type of media a specialized data format. This approach is implemented, for example, in the Distributed Multiparty Desktop Conferencing System (MERMAID) [29]. Video, voice and data are transmitted through 2 B-channels of ISDN. The problem is that ISDN B-channel bandwidth is not enough for transmitting clear video images in real-time and the delay is too large. Another system is the Bellcore Integrated Media Architecture Laboratory's (IMAL) system, which they call Computer Supported Cooperative Work (CSCW) [15]. CSCW is concerned with internetworking and integration of networked CAD/CAE resources and networked facilities for multi-media teleconferencing and messaging in a multimedia environment. CSCW is implemented on three different networks. Control data and text is transmitted over a LAN, video transport is provided by analog coaxial cable carrying NTSC signals, and audio transport is provided by analog balanced-line low-impedance cabling [15]. The CCITT Expert Group on Visual Telephony's development increases the number of B-channels to from 23 to 30. Study group XV addressed the problem and produced CCITT Recommendation H.261: "Video Codec for Audiovisual Services at px64 kbits". [5] They have also done work in the broadband ISDN ("Integrated Services Digital Network") [26][27][1][32] community, where the conceptual value of integrated services is accepted, but the integration (multiplexing) of the media streams comes in lower (ATM) layer.

Towards an *integrated approach for network transmission* there are efforts by the MPEG (Moving Picture Experts Group). The activities cover video compression, associated audio compression, audio-video synchronization and the issue of synchronization and multiplexing of multiple compressed audio and video bit streams [5]. Ziegler and Weiss have experimented with integrated approaches and developed some mechanisms for integrated voice and data conferencing [31]. Their goal was to transmit real-time voice and non-real-time text data. They proposed the "shuttle packet" method, where they used a data format which could take either voice or text. Their proposal used the same data format for both media, but they transmitted only one data type at a time, i.e., either the shuttle packet had voice or it had text. Differentiation was done with a control flag in the data unit[31].

Little and Ghafoor's work[12][13][3][4] on integrated multimedia messages, composition and synchronization is somewhat similar to our protocol work, although it differs in the major respect that we preserve composition and synchronization properties end-to-end (in our model, application-to-application) in the networked multimedia system.

1.2 Organization of this paper

This paper is organized as follows. Section 2 describes the assumptions for our multimedia model and the integrated media approach for dealing with multimedia in real-time distributed applications. It also enumerates some research problems connected with multimedia real-time networking, such as multiple quality of service (QOS) parameters and synchronization of streams of sensory (i.e., real-time) data.

Section 3 focuses on the interchange multimedia message format. The IMM packet is the focus of media integration. Our IMM format's novelty stems from the generality of the header format, which has the function of a temporal relation descriptor. The IMM structure can express the intermessage and intramessage dependencies of a very general multimedia message.

We also propose a composition protocol for generation of IMM and decomposition protocol for presentation and synchronization of IMM, and a QOS handler. The QOS handler is built on the real-time multimedia protocols. This handler is a service entity in the application layer, which supports varied QOS, and maps end-user QOS requirements to system capabilities. It provides dynamic control of QOS to the user.

Section 4 provides a short overview of the *U-Penn Experimental Multimedia Conferencing System* (UPEMMCS) project. Among the research goals of the project is an experimental implementation and evaluation of the integrated media approach.

In Section 5 we offer some closing thoughts on the integrated approach and point to future work.

2 Integrated Multimedia Model

2.1 Our Assumptions

We make the following assumptions about our operating environment and the capabilities it provides:

1. Each workstation has some set of devices for capture and reproduction of Audio, Video, Text, Graphics, etc. These devices, the *sources* and *sinks* of media, serve to create the multimedia environment at that network endpoint.

2. Real-time transport, high-bandwidth and a global clock (e.g., the Network Time Protocol or some functional equivalent) are available.

3. The only delays which are fixed and lower-bounded are transmission and propagation delays.

4. Synchronization is embedded in the application layer of the Open System Interconnection (OSI) model or in the orchestration layer of the multimedia architecture[7]. Embedding the synchronization of the IMM's in the application layer should speed up the presentation, reduce the need for buffering and resynchronization processes in the network, and hence increase the service capacity available to applications.

5. The time needed for local manipulation of data, e.g. compression, decompression, noise and error handling, etc., is small or can be made small by augmenting the workstations with additional hardware.

6. Resource allocation tuning knobs (such as bandwidth reservation, allocation and management) are available [34].
7. The system behavior for distributed multimedia applications can be decomposed according to a client/server model.

2.2 Integrated Approach

With available high-speed networks and advanced hardware technology in the workstations as we pointed out earlier, it becomes possible to use many media concurrently. All these communication media appear to a user (man or machine) as a "sensory environment" within which communication with other users is embedded. Thus, it seems sensible from the user point of view to send to other users the whole environment (the "sensory context") associated with a particular time interval. From the networking point of view it means to capture the heterogeneous multimedia environment and send it in integrated form as a data unit to the network. The available bandwidth of advanced networks, coupled with compression techniques, support transport of such quantities of data. The integration and disintegration of the multimedia environment is embedded in the application layer because it should map the user's demands and provide simultaneous control over the environment and QOS.

This approach is attractive both conceptually (because it views the multimedia system's task as reproducing an environment) and technically (synchronization is eased because related data stay together). We see some strong arguments in favor of the integrated approach:

- data which are created together stay together (in B-ISDN created multimedia data are separated into homogeneous media streams),
- synchronization software tends to be much simpler, and once an IMM message is created, the same relationship is observed at each point where the data is displayed, (in B-ISDN the synchronization is more complicated because of unequal delays for separate data streams belonging to one environment),
- it is easier to add new services and media,
- it is easier to record data for later playback, because the IMM has a dependency descriptor (Header of IMM) of the different media included in it. By recording the data the dependency descriptor is stored together with the data belonging to the environment and by later playback we can easily reproduce the environment.
- multicasting is considerably easier, since customization is done by *clients* rather than multicast *servers*.
- multiple application-specific solutions need not be developed for common problems, e.g. privacy. If data is private, related data is generally private as well, and encipherment of such data should be done end-to-end. Medical images and physician's voice annotations to such images are likely to be equally sensitive.

Using the concept of distributing sensory environments, we designed the IMM format described in Section 3.1.

2.3 Problems, or why there's no free lunch

As a consequence of the integrated approach many research problems emerge. Among the most difficult problems are the *heterogeneity of end nodes* and the *heterogeneity of media* Quality of Service *(QOS) requirements.* End nodes typically consist of a computer and a number of sensory devices as input and output devices such as displays, microphones, cameras and CD-players. The networking QOS requirements often enumerated[18] are:

1. *Delay-Sensitivity*

   To achieve the real-time constraints on audio transmission, we must have bounded transmission delays (so that reasonable conversations can be held) and bounded interarrival delays for consecutive packets. The latter delay can be varied against packet size to guarantee continuous audio. Intermedia delays are also important (e.g., to prevent loss of lip-synchronization, audio packets should not be delayed relative to video packets).

2. *Loss-Rate-Sensitivity*

   Loss-Rate sensitive media are text, and to a lesser degree, audio. In the case of video the eye can extrapolate the missing frames more effectively than the ear can "restore" missing audio packets or characters[1].

3. *Bandwidth Allocation-Throughput*

   Some transmitted media have high request on the network bandwidth. As an example consider video source with 30 frames/sec., with resolution 640x480 pixels and 8 bits/pixel. For an uncompressed video source the bandwidth requirement is 72 Mbps.

¿From the user point of view the QOS requirements might be

1. *Image Resolution,*
2. *Sound Quality,*
3. *Text Quality,*
4. *Price for Quality.*

There is a direct mapping from the user QOS requirements to the networking QOS requirements and vice versa. This raises questions not only on how to deal with heterogeneous QOS requirements, but also how to allow dynamic changes of user QOS requirements. Dynamic changes might be needed where the application desires change. For example, an user might want to rearrange the presentation of media on a display, with image quality made proportionally larger as the image size increases; another possibility is reducing the required signal to noise ratio for audio as the application lowers the volume.

The next problem is the *synchronization* of different media on the receiver side. Our integrated approach makes it easier because created multimedia stays together during transmission and possibly archival; skewed intermedia delays vanish as a problem.

Interchange Message:

Envelope	Content

Content:

Header	Body

Body:

Media Param.	Text	Media Param.	Video	Media Param.	Voice	...

Figure 1: *Integrated Multimedia Message*

3 Data Format and Protocols

3.1 Integrated Multimedia Message

To transmit the multimedia environment, the multimedia data must be captured and prepared for transmission. We focus our attention on the step where disparate sources of traffic are combined into a message. Here, we assume that the "message" is the unit of data acceptable to lower layers of whatever protocol hierarchy the real-time multimedia application uses.

The integrated multimedia message format is imposed on a message which contains discrete media such as text and graphics, with interelement sequencing but lacking interelement timing, as well as continuous media such as video and voice, which add interelement timing requirements. The time dependency of all information types in the IMM as stated before is obvious.

We can combine discrete and continuous media, since while the continuous media appear to listeners or viewers as continuously changing over time, their internal representation in a digital system is discrete. It consists of single audio samples or video frames. Each IMM thus is associated with a time interval in the time scale of the application. The time intervals are not assumed to be of fixed length, although this may be the most common case. The duration could be variable, depending on the properties of data comming from the devices such as camera and microphone. For example, long periods of silence or inactivity can be taken advantage of by not sending messages as frequently. However, in any case, any structural relationship such as sequencing must be preserved.

Our IMM is illustrated in Figure 1. The IMM contains a general descriptive *Envelope*, which includes the message addressing parameters and parameters for the intermessage synchronization.

The *Content* includes the general *Header* and the actual multimedia data in the *Body*. The general header can be viewed as temporal relation descriptor which has the information for the intramessage synchronization. *Media Parameters* in the body are important for the presentation of each media, or for the particular application, the media has to be delivered to. A summary table of parameters and brief description for each is given in Table 1. In essence, the sender's data are multiplexed into a single multimedia stream composed of IMM's and demultiplexed on the receiver side for presentation or computation.

Structure-Elements	**Functionality**
Envelope	descriptive and addressing information
Receiver	receiver address in the application layer
Sender	sender address
Message Identifier	unique identifier of the IMM
Message Type	in dependence of QOS (loss-rate sensitive data) divide the message if it is original or copy IMM
Time Begin	beginning of the time interval, the IMM was created
Time Duration	length of the time interval, the IMM belongs to
Content	contains all information the application layer needs to provide the processing at the receiver side
Header	information related to the local handling of IMM
Body Description	description of the body (single, composed)
Single Body	specifies the information type, because the body has only one kind of information
Composed Body	specifies the body parts in the body
Body Parts Relations	relations among the body parts in the particular time interval (independent, synchronized)
QOS-maxima	QOS parameters for every media of the sender
Body	contains the actual content
Media Parameter	specifies the type, length, time parameters of the body part
Identifier	unique ordering number of the body part inside of the IMM
Time Begin	time when the body part was created
Time Duration	duration time of the body part
Optimization of Data	media specific processing (e.g. compression)
Data	data with specific descriptions of layout, etc. according to application - specific standards

Table 1: *Structure and Functionality of IMM*

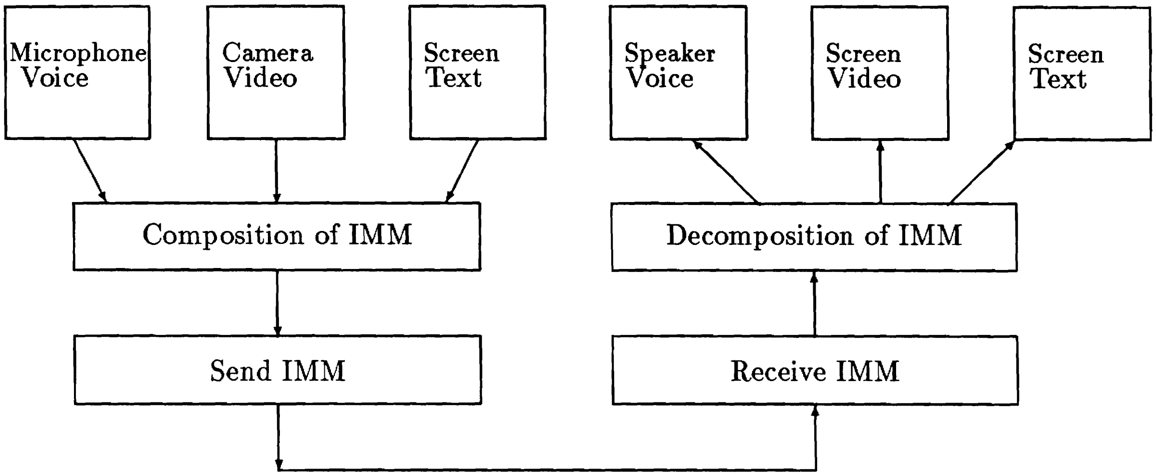

Figure 2: *Client-Client Protocol*

3.2 Multimedia Protocols

We focus in this part on two important protocols, the *composition protocol* and the *decomposition protocol*, which are parts of the Client-Client protocol shown in Figure 2.

3.2.1 Composition Protocol

The composition protocol implements the composition service of the application layer. Different media are captured in real-time, integrated for a given time interval into an IMM packet and sent to the network. The protocol service primitives are

- *Composition-Connection-Establishment*,
- *Composition-Transfer*,
- *Composition-Connection-Release.*

In **composition-connection-establishment** all processes from Figure 3, which participate in the composition-transfer, are created by the *Send Manager.* In this set-up phase, all processes are passed information needed for communication such as receiver addresses and the QOS parameters.

Composition-transfer is carried out among components as *I_Audio Manager, I_Video Manager, I_Text Manager*, which sense different media, *Composition*, which gathers the captured data and creates the IMM, and *Send-To-Net* component, which passes the IMM to the network. If the composition component must process loss-rate sensitive data, it adds redundancy to the IMM for these data data. This trades bandwidth for reliability, a good tradeoff in real-time systems where retransmission is difficult, if not possible. All processes are periodic real-time processes, although the media managers, composition, and send-to-net components may have different periods. The periods for all processes depend on the quality of service and the resources the client machine has available. Event communication in composition transfer is shown in Figure 3.

If the user or the network wants to close a connection, the **composition-connection-release** primitive is invoked. This deallocates used resources and participating processes terminated.

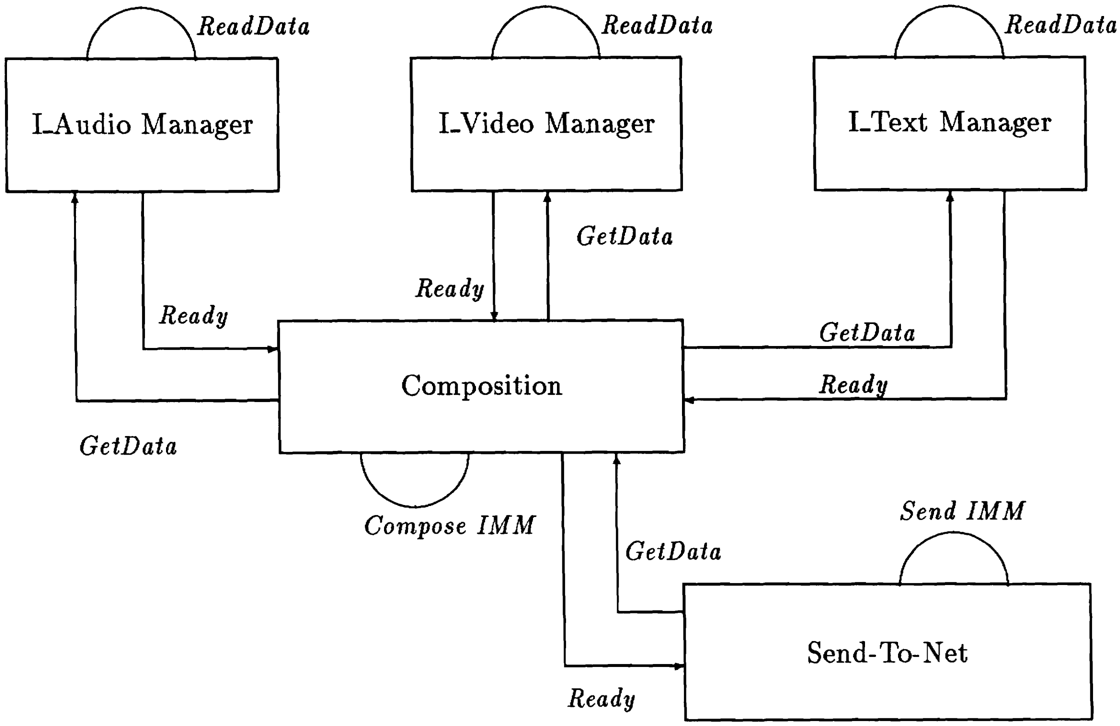

Figure 3: *Composition-Transfer*

3.2.2 Decomposition Protocol

The decomposition protocol includes functionality such as *receiving* the IMM stream, *decomposition* of the IMM's, and *output* to the user interface. For more complex applications, output may be redirected to various sub-programs which comprise the application. Intermedia and intramedium *synchronization* of the IMM's body parts must also be performed.

Analogous with the composition protocol, the **decomposition protocol** implements the following service primitives:

1. *Decomposition-Connection-Establishment*,
2. *Decomposition-Transfer*,
3. *Decomposition-Connection-Release.*

In the **decomposition-connection-establishment** all processes shown in Figure 4 are created by the *Receive Manager.* The manager also delivers user QOS capabilities to the decomposition component and media managers.

Functionality of the components during the **decomposition-transfer** phase is described in Table 2 and event communication is presented in Figure 4; note the addition of synchronization information flow from Figure 3.

The **decomposition-connection-release** primitive will be activated if the connection to remote users is closed. Resources are deallocated and participating processes terminated .

Component	Functionality
Recv-From-Net	receives the IMM stream, where the IMM's are not implicit ordered. This component has to preserve the ordering, unique existence of the IMM with respect to the delay requirements and the buffering of the IMM's for the intermessage synchronization. If one of the IMM's (original or copy) arrive in the given delay, the later (obsolete) IMM will be deleted.
Decomposition	processes the header and decomposes the body.
Output Media Manager	processes the media parameters and provides intramessage synchronization for output and presentation of the single data.

Table 2: *Functionality of the Components*

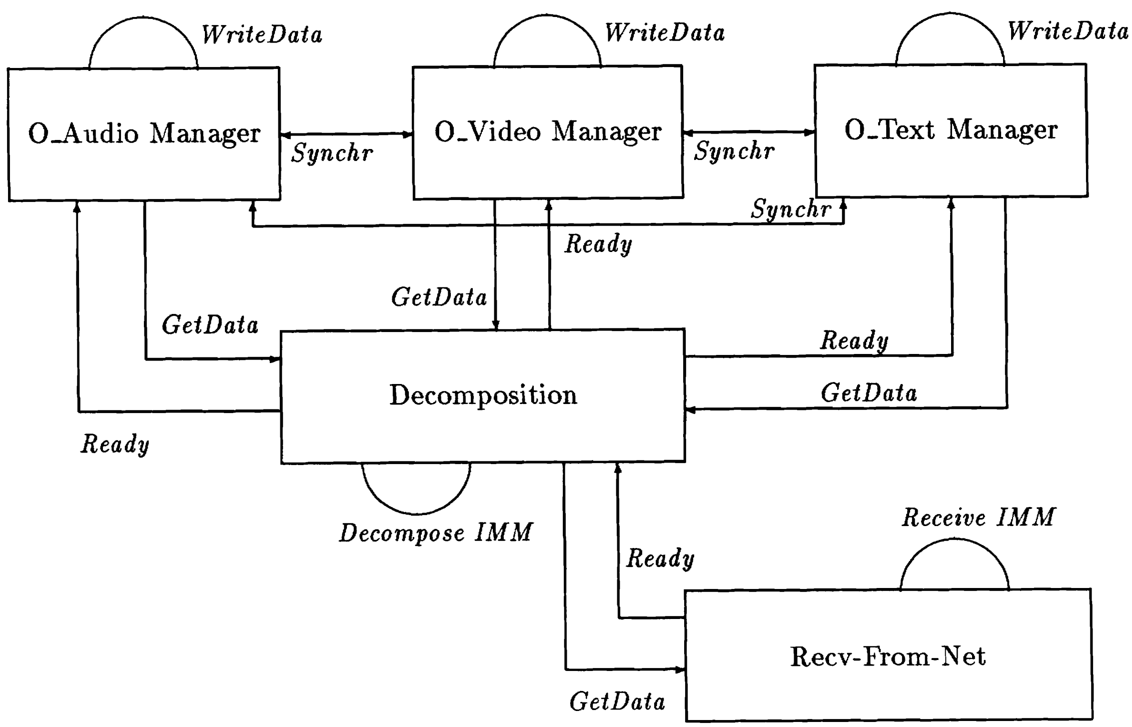

Figure 4: *Decomposition-Transfer*

3.3 Quality of Service Handler

The QOS handler is a service entity which provides a dynamic mapping of different quality of services for each medium from users to the network.

The QOS handler is comprised of a *user part* and a *networking part.*

In the user part the user has the opportunity to customize his or her QOS parameters. This customization process (which can occur at any time) includes such choices as: sound quality (e.g., Telephone or CD Quality), video quality (e.g., Resolution, Color, Bits/pixel, Exposure Time) as well as error rates, delays and price-sensitivity if there are charges.

The networking part maps the user requirements to the networking QOS provision capabilities, e.g., media throughputs, delays, loss-rate, etc. With this data the QOS handler can communicate with resource allocation schemes[2] to get the required bandwidth and other resources, or discover it is unavailable. The networking part also sends: user QOS parameters such as resolution of video, which media are loss-rate sensitive, and what is an acceptable delay for the medium, *etc.* This data is communicated to the composition and decomposition component of the particular protocol.

4 The UPEMMCS - A Brief Overview

We are designing and implementing an experimental multimedia conferencing system. Our intent is to gain experience and to have an apparatus with which we perform experiments on the AURORA network. The issues we are currently attempting to study and understand are:

1. an integrated format for various information types as described in the earlier sections of this paper, such as continuous media types (video and audio), discontinuous media types (text), and media expressed in instruction form (graphics or commands for telerobotic devices).

2. synchronization of various information types which have timing properties (as is the case with sensory data), and the impact of composition and decomposition on these information types.

3. requirements for network protocols for real-time applications, especially those with distinct QOS measures.

4. hardware support necessary to augment workstations in real-time networked multimedia systems.

5. dynamic QOS parameters, i.e. how the receiving user might have control over the quality of the services (such as sound, video, and text) the end-system should provide.

Since we have described the details of UPEMMCS[25] elsewhere, we will focus here on the questions UPEMMCS will help us answer. In particular, we want to create an environment for testing the integrated media approach in an advanced networking system.

We have decomposed the end-nodes in *Layered Approach Multimedia Provision* (LAMP) as illustrated in Figure 5.

UPEMMCS is based on the LAMP model, and the work in this paper has focused on the user interface and media services areas. We are prototyping the user interface software on the NeXT workstations, and the media services on SUN workstations. The driving application used for the user interface research is a student-teacher conference. The user interface results will be used to develop a second generation user interface portable across NeXT, SUN, and IBM RS/6000 workstations.

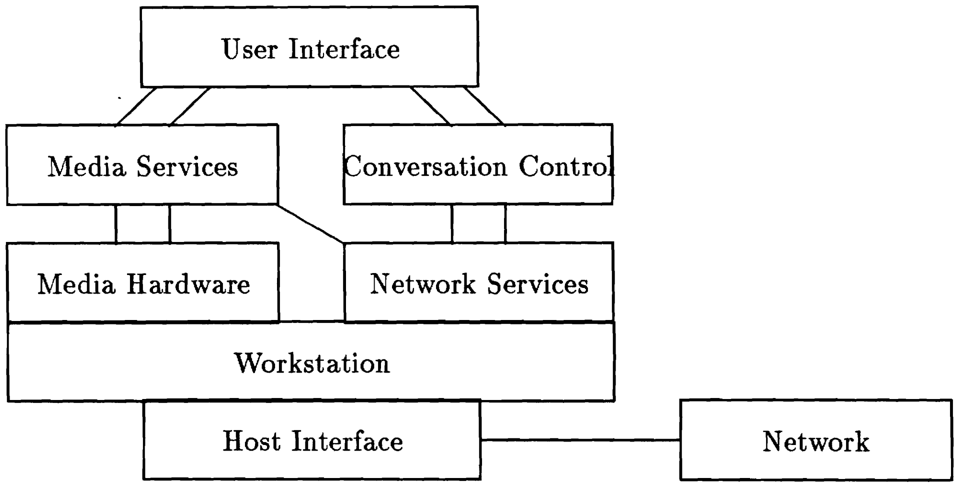

Figure 5: *LAMP Model*

SUN workstations with existing hardware support, such as VideoPix and the AM79C30A Digital Subscriber Controller Chip for the basis for implementation of the composition, decomposition protocols and QOS handler as described in Section 3. This is now being implemented, and should be portable except for minor device details and some reliance on TCP/IP services.

The RS/6000 is the LAMP hardware development platform, and as of the date of this paper (01/ 1992), we have implemented and tested an audio CODEC, a video capture board, a 100+ Mbps DES cryptographic board, and a 130+ Mbps host interface board for ATM-based networks. All of these devices are connected to the RS/6000 through the Micro Channel I/O bus. We can perform card-to-card data transfers at very high speeds, offering the possibility of hardware optimizations of the IMM packetization scheme. There are several technical advantages to developing our own multimedia workstation hardware on a platform distinct from the software development platforms:

1. hardware support for some media is currently available for the SUN and NeXT stations
2. we force the software solutions to be portable
3. development efforts can proceed in parallel, allowing crosstalk and learning as well as more rapid progress
4. support for (or tolerance of?!) heterogeneity must be *designed* into the entire architecture, rather than retrofitted
5. we have an increased opportunity to address issues of scale, to which we have alluded in this paper

The integrated multimedia message is used as an interchange message format, and the multimedia protocols (as described in Section 3) are embedded in the "Media Services" portion of the LAMP model. Media Services for UPEMMCS divide the functionality between **Multimedia User Agents** and **Multimedia System Agents**. **Multimedia User Agents** provide the local services such as generating, sending, receiving and synchronizing multimedia stream. **Multimedia System Agents** (which implicitly communicate with the "Network Services" portion) provide multicasting of the multimedia stream to every conference participant. UPEMMCS is embedded in the application layer of the networking architecture stack, as shown in Figure 6.

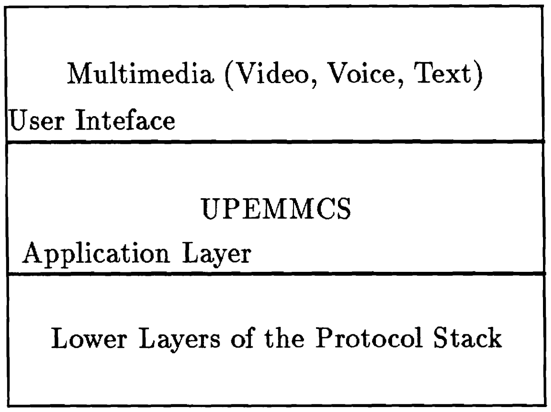

Figure 6: *Layer View of UPEMMCS*

We have made some assumptions for our implementation efforts. In particular, the user QOS parameters such as resolution, frames/sec, bit/pixel, throughput, *etc.,* are fixed by hardware capabilities. We have used TCP/IP for transport due to its ubiquity. We have adopted a "wait-and-see" attitude towards the transport protocol development efforts designed to deal with media, and intend to adopt whatever results research proves effective. With the use of TCP/IP, we adopt by default a connection-oriented reliable transport protocol.

Note that our use of connection-oriented semantics does not imply two-way connectivity. A "connection" is a unidirectional information flow between a sender and a receiver - the sender in any connection should specify which services it is willing to deliver (typically a proper subset of what it is *capable* of delivering) to a receiver. The sender embeds these choices in the IMM packets which it puts onto the network; the receiver then decides, based on the information in the IMM packet, what it is capable of reproducing, and what it is willing to reproduce. To instantiate a two-way connection, another connection is established reversing the role of sender and receiver.

This is not to say that conversations cannot be customized; out-of-band control information can always be used to augment the multimedia information system; for example, cryptographic key exchange can be used to allow IMM packets with privacy-protected data to be examined. The key ideas of this "dissemination-oriented"[33] model are:

1. multicast and broadcast should be easy;

2. control should be as distributed as possible (this has advantages in both failure tolerance and scaling; and

3. heterogeneity is a reality, and must be addressed.

Multicast support is extremely desirable, and the model for multicast must scale effectively. Many of the important applications for multimedia demand multicast:

- distributed classrooms, where an educator lectures to several remote sites, and questions from remote sites should be reproduced for all students

- multiparty conversations, where, for example more than two interacting parties exist. A particularly telling example for us is the AURORA collaboration, where four or more parties must meet to reach consensus on actions and research agendas; this usually involves at least one plane flight. Here, there should be multicasts from each sender to the rest of the group, and the focus of each receivers interest may differ.

5 Conclusions and Future Work

It is a bit premature to claim that the integrated approach to networked multimedia is comparatively better or worse than one approach or another. In this paper, we have argued some of the merits of the approach, tackled a few of the technical problems with proposed solutions such as the descriptive headers used for IMM packets, and shown that many QOS difficulties can be addressed by combining other mechanisms with real-time data delivery. Using our proposal for IMM packets, we have given protocols for composition and decomposition of these packets into sensory environments.

We intend to verify our hypotheses experimentally in the context of the experimental setup described in Chapter 4 of the paper. We see a number of important open problems that we have discovered in our research:

1. Quality of Service measures must be enumerated, categorized, and prioritized. For example, we could enumerate reliability and real-time, categorize them as non-time-dependent and time dependent, respectively. We expect they could be prioritized with real-time given a higher priority, since reliability mechanisms can be built using high bandwidth real-time transport, but the reverse is not true.

2. A way of precisely formulating relations between QOS measures is highly desirable; this would allow IMM packets to have a QOS descriptor attached which is derived from the QOS measures of its constituents.

3. Issues of scalability must be tested. Most research multimedia systems connect at most a few dozen sites, which are typically geographically co-located offices or work spaces. We have discussed scaling issues in our advocacy of dissemination-oriented[33] multimedia communication. The evidence of Radio, Broadcast and cable TV, and recorded media shows that

   (a) most media use is not interactive;

   (b) people are generally interested in having a large number of possible selections available;

   (c) incredibly large scales are possible. We should imagine multimedia systems on the scale of the Internet, and design for them.

4. Heterogeneity in end-nodes should be accommodated; while we have made some proposals in this paper, this remains a general problem and needs a precise solution. There is hope; black and white televisions can display sound and image from color recorded video. While this might properly be considered an issue of scale, in fact it serves to test the generality and flexibility of any design proposal.

5. Human-factors research is a major source of questions for our community. In particular, such research helps to answer questions about utility and feature selection, and may help in offering practical input on QOS tradeoffs. For example, here is a technical question which requires psychological measurement: if I have a total of B bits of video capacity (B bits per second of bandwidth...), and the product of the number of color planes (c) and the number of displayed pixels (p) is constrained by B, how can I optimize my choices of c and p? Do the tradeoffs change as B increases? How do application characteristics affect the tradeoffs (e.g., journal publishers might wish to emphasize p, advertising agency artists c, and photography magazine editors will never be satisfied with any value of B!)?

Among the most exciting directions to pursue is the possible generalization of our approach from our current conception (that of multiple media) to what might be best described as "mixed information" systems. "Mixed information" systems would use a generalization of the integrated multimedia approach we have proposed in this paper to deal with many problems where sensory information must be combined with information types used for command and control. We alluded, for example, to the use of our integrated media format for the sensory apparati used by telerobotics applications; a mixed information model might view the communication as consisting of a complex two-way information flow between remotely located intelligent instruments and a managing node. The robot might want to mix gathered sensory information with complex real-time database queries to send to the managing node, while replies might consist of commands to perform joint motions and digital images for object recognition pattern-matching. Such mixed information systems allow high-speed networks to service a much broader range of applications and further improve our lives.

References

[1] Hans Eriksson, "Multimedia in MultiG", *Proceedings of the 1st MultiG Workshop Stockholm*, November 1990.

[2] L. Zhang, "A New Architecture for Packet Switching Network Protocols", Ph.D. Thesis, MIT, July 1989.

[3] Thomas D.C. Little, Arif Ghafoor, "Multimedia Synchronization Protocols for Broadband Integrated Services", *IEEE Journal on Selected Areas in Communications*, 9(9), pp. 1368-1382, December 1991.

[4] Thomas D.C. Little, Arif Ghafoor, "Spatio-Temporal Composition of Distributed Multimedia Objects for Value-Added Networks", *IEEE Computer Magazine*, October 1991.

[5] Didier Le Gall, "MPEG: A Video Compression Standards for Multimedia Applications", *Communications of the ACM*, 34(4), April 1991.

[6] A. A. Lazar, A. T. Temple, R. Gidron, "An Architecture for Integrated Networks that Guarantees Quality of Service", *International Journal of Digital and Analog Communication Systems*, V. 3, pp. 229-238, April-June 1990.

[7] C. Nicolaou, "Architecture for Real-Time Multimedia Communication System", *IEEE Journal on Selected Areas in Communications*, 8(3), pp. 391-400, April 1990.

[8] R. Steinmetz, "Synchronization Properties in Multimedia Systems", *IEEE Journal on Selected Areas in Communications*, 8(3), pp. 401-412, April 1990.

[9] *Digital Eye User Manual*, Metaresearch, Inc., 1990.

[10] R.G. Herrtwich, "Time Capsules : An Abstraction for Access to Continuous Media Data", *IEEE Real-Time Systems Symposium, Proceedings*, 1990.

[11] *NeXT Technical Documentation Release 0.9*, The Publication Group at NeXT, Inc., May 1989.

[12] T.D.C. Little, A. Ghafoor, "Synchronization Properties and Storage Models for Multimedia Objects", *IEEE Journal On Selected Areas in Communications*, 8(3), pp. 413-427, April 1990.

[13] T.D.C. Little, A. Ghafoor, "Network Considerations for Distributed Multimedia Object Composition and Communication", *IEEE Network Magazine*, 4(6), pp. 32-49, November 1990.

[14] *VideoPix Manual*, Microsoft, 1990.

[15] Lester F. Ludwig, "Integration of CAD/CAE with Multimedia Teleconferencing and Messaging via Broadband Networks and Shared Resource Servers", *Proceedings of the First International Conference on System Integrations*, April 1990.

[16] G.M. Woodruff, R. Kositpaiboon, "Multimedia Traffic Management Principles for guaranteed ATM Network Performance", *IEEE Journal on selected Areas in Communications*, 8(3), April 1990.

[17] Ching-Hua Chow, Motomitsu, "Achieving Multimedia Communication on Heterogeneous Networks", *IEEE Journal on Selected Areas in Communications*, 8(3), April 1990.

[18] David C. Feldmeier, "QOS: Requirements and Architecture, End System Considerations for Quality of Service", *Technical Report, Bellcore*, October 1991

[19] D. Comer, "Internetworking with TCP/IP: Principles, Protocols and Architectures", Prentice-Hall, 1988.

[20] D. D. Clark, B. S. Davie, D. J. Farber, I. S. Gopal, B. K. Kadaba, W. D. Sincoskie, J. M. Smith, D. L. Tennenhouse, "The AURORA Gigabit Testbed", *Computer Networks and ISDN Systems*, (to appear).

[21] Gerd Schurmann, Uwe Holzmann-Kaiser, "Distributed Multimedia Information Handling and Processing" *IEEE Network Magazine*, pp. 23-31,November 1990.

[22] J. S. Sventek, "An Architecture Supporting Multimedia Integration", *Proceedings, Office Automation Symposium*, pp. 46-56, IEEE CS Press, 1987.

[23] M. E. Hodges, R. M. Sasnett, M. S. Ackerman, "A Construction Set for Multimedia Applications", 1989.

[24] J. J. Garcia-Luna-Aceves, "Towards Computer-Based Multimedia Information Systems", *Computer Message System-85*, R. Uhlig (ed.), Elsevier/North-Holland, pp. 61-77, 1985.

[25] K. Nahrstedt, R. Sajja, C.B. S. Traw, S. Udani, J. M. Smith, "LAMP: Layered Approach to Multimedia Provision", *DSL Technical Report*, submitted for publication, 1992.

[26] S. E. Minzner, "Broadband ISDN and Asynchronous Transfer Mode (ATM)", *IEEE Communications*, Vol. 27, pp. 17-24, 1989.

[27] R. Ballert, Y. C. Ching, "SONET: Now It's the Standard Optical Network", *IEEE Communications*, Vol. 27, pp. 8–15, 1989.

[28] R. Cordes, T. Kummer, "Multimedia Communication and Information Management Based on Available and Emerging Standards", *Technical Report TELENORMA*, November 1991

[29] K.Watabe, S.Sakata, K. Maeno, T. Ohmori, "Distributed Desktop Conferencing System with Multiuser Multimedia Interface", *IEEE Journal On Selected Areas in Communications*, May 1991.

[30] Luis-Felipe Cabrera, Darrel D.E. Long, "Exploiting Multiple I/O Streams to provide High Data Rates", *Conference Proceedings, Multimedia-For Now and the Future, Nashville, Tennessee*, June, 1991.

[31] C. Ziegler, G. Weiss, "Mechanisms for Integrated Voice and Data Conferencing", *SIGCOMM '90 Symposium, Philadephia*, PA, September 1990

[32] Masatoshi Kawarasaki, Bijan Jabbari, "B-ISDN Architecture and Protocol", *IEEE Journal on Selected Areas in Communications*, 9(9), pp. 1405-1415, December 1991.

[33] D. R. Cheriton, D. Skeen, "Dissemination-Oriented Communication Systems", Stanford Computer Science Technical Report, 1992 (to appear).

[34] D.P. Anderson, S.Y. Tzou, R. Wahbe, R. Govindan, M. Andrews, "Support for Continuous Media in the DASH System", *Proceedings, The 10th International Conference on Distributed Computing Systems, Paris, France*, May 1990.