An Application-Driven Approach to Networked Multimedia Systems

Klara Nahrstedt Jonathan M. Smith*
Distributed Systems Lab., University of Pennsylvania, Philadelphia, PA 19104-6389

Abstract

Among the most difficult challenges in providing real-time multimedia services[9] to applications are coping with heterogeneity of end nodes (consisting of various input and output devices) and heterogeneity of media Quality of Service *(QoS) requirements[11].*

We discuss several architectural choices possible in logical multiplexing, accommodating QoS dynamics, and functional divisions between senders and receivers. Our application requirements favor integration rather than channelization, supporting QoS dynamics with a renegotiation mechanism, and pushing complexity towards receivers.

We are testing the choices via implementation of a telerobotics/teleoperation application. This application is particularly challenging due to its demand for multiple concurrent media streams with varied QoS requirements.

1 Introduction

Modern high-speed computer communications networks[13] allow multiple high-bandwidth media applications on a single network fabric. These new networks are also characterized by their flexibility – the combination of packet-switching and intelligent terminal equipment permits considerable customization of a channel. Customizations might include allocated bandwidths and controlled interarrival delays.

Communication media appear to a user (man or machine) as a "sensory environment", within which communication with other users is achieved. To us, it seems sensible to communicate to other users the whole environment (the "sensory context") associated with a particular time interval. In a networked multimedia system, propagating the sensory environment in some integrated fashion through the network raises some interesting system design questions.

Integrated means that media share a common means of transport, and are viewed as a composite by the communicating portions of a distributed application. The means of transport may be *logically* channelized within the internals of the system to reflect differences in application QoS requirements, but the multimedia data are treated as a *composite* by both sending and receiving applications, rather than individually.

In this paper we focus on an architecture for networked systems which provides propagation of an *integrated* sensory environment (consisting of real-time multimedia) through a network. As a consequence of the way we have partitioned control, the architecture flexibly copes with future developments, e.g, new requirements, media, and devices. Flexibility is particularly important in long-lived systems, and we see a number of important application areas for non-traditional "media" (such as tactile data), which should be accommodated by a "multimedia" system.

1.1 Approaches to Multimedia Networking

Real-time multimedia networked systems differ in their organization of data and data transmission approaches.

One approach (the *multi-channel approach*) transmits different media in separate streams (channels) and has for each type of media a specialized data format, and specialized protocols. The integration of the sensory environment is done only at the user interface. In this case, the technical challenge appears to lie in synchronization at the receiver. Preserving timing relations among the media can be hard, since some information about the intermedia relation gets lost.

A second way (the *interleaving approach*) to propagate the integrated sensory environment through the

*Research support for this work came from Bellcore (through Project DAWN), and by the National Science Foundation and the Advanced Research Projects Agency under Cooperative Agreement NCR-8919038 with the Corporation for National Research Initiatives. Prof. Smith was also partially supported by an IBM Faculty Development Award.

network is through time-multiplexing (interleaving) of some media over a single connection. Control information (a *descriptor*) is prepended to integrated packet - it includes information about the media components and the intermedia relations. The technical challenges in this case are that the packets can be unwieldy and the QoS parameters for integrated packets may reflect the worst-case QoS parameters (e.g., real-time demands) for each of the constituent media.

A third approach (the *application-driven approach*), which is a hybrid of the other two with some additional services, provides a mechanism during call establishment and call transmission to guarantee that the sensory environment is recreated at the receiver side. The approach provides an application-specific combination of multi-channel approach and interleaving approach – the driving forces are the application requirements, which then decide which approach is mapped to which set of media. This approach requires support mechanisms in the form of new services *negotiation* and *renegotiation*, which take the application requirements and communicate them to subsystems which guarantee the application requirements. Renegotiated QoS values allow the dynamics of the application to be more precisely reflected in its demands on the network, and allow the network resources to be better managed through cooperation with applications[7].

1.2 Related Work

The multi-channel approach is implemented in the Distributed Multiparty Desktop Conferencing System (MERMAID) [14]. Video, voice and data are transmitted using two ISDN B-channels. Unfortunately, the ISDN B-channel bandwidth is insufficient for transmitting clear video images in real-time and the delay is unacceptable for many applications. Another multi-channel system is the Bellcore Integrated Media Architecture Laboratory's (IMAL) system, called Computer Supported Cooperative Work (CSCW) [10]. CSCW is concerned with internetworking and integration of networked CAD/CAE resources, as well as networked facilities for multi-media teleconferencing and messaging, embedded in a multimedia environment. CSCW is implemented using three different networks. Control data and text data are 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 [10].

There are efforts towards an interleaving approach for network transmission 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 [8]. Ziegler and Weiss have experimented with integrated approaches and developed some mechanisms for integrated voice and data conferencing [15]. 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[15].

The application-driven approach is based on an extensive body of work on real-time transport protocols, and architectures [6] for achieving real-time goals such as low jitter [5] and low delay [3]. Some negotiation protocols are proposed in the ISO 95 work, but application-to-network communication appears problematic due to the strictness of the OSI peer-to-peer architecture. Jung and Seret [2] have proposed a translation scheme for layer-to-layer communication, but have not exploited their scheme in either a broader distributed applications context or an experimental setting.

1.3 Organization of the Paper

The description of our 'integrated' approach, which uses negotiation/ renegotiation, as well as activity scheduling based on QoS, is given in Section 2.

The proposed approach is explored in a telerobotics application. The architecture, expected results and implementation specific issues are presented in Section 3. Section 4, which concludes the paper, summarizes what we now know and what remains to be understood.

2 Application-driven Approach

To recreate an integrated sensory environment at the receivers, we use the application-driven approach outlined above. We want to provide a mechanism during call establishment and data transmission which guarantees the recreation of the sensory environment. The environment consists of media *streams* with different timing, error, and bandwidth requirements, as possibly periodic delivery. To recreate the environment, we provide two major components in the network architecture:

- negotiation and renegotiation of quality of service(s) during the call establishment phase. It includes negotiation/renegotiation of application requirements for the integrated sensory environment, translation of these requirements into network requirements for one or more partial connections (per medium), and negotiation/renegotiation of network requirements/guarantees using an admission control mechanism.

- transport of the streams through different negotiated QoS connections; necessary periodic scheduling based on QoS is arranged during the call transmission phase.

While here we assume that the sender creates a single integrated sensory environment, our approach can be extended to the case when the sources are distributed by altering the negotiation process.

2.1 QoS Model

The first task is to describe the QoS parameters which specify the objects representing the networked multimedia system. Our model is shown in Figure 1[1].

End-point objects are represented as **input** and **output** devices. These objects are described with application QoS parameters, which actually represent the application requirements for the sensory environment. We split QoS into two categories. **Application QoS** is "quality" in terms meaningful to application services, i.e., how well the application can recreate data which satisfy the expectations of end users. Specification is in terms of **application characteristics**. The application characteristics parameters include information on the multimedia **stream description** and the **media relations**, such as communications topologies and entity roles (e.g., synchronization, any necessary media conversion, and whether an interleaving/multichannel variant is required for transport). The stream description maintains **media quality** parameters. Some example parameters for quantized continuous media are the *sample rate*, *sample size*, *compression algorithms* and *sample loss rate*.

The transmission object is the **network**, which is characterized by **network QoS parameters**, such as **throughput pledge** (allocated bandwidth) as well as other **performance requirements** (such as delay bounds, loss rate). For the network QoS parameter set, we have found the parameters used in the Berkeley Tenet protocol suite[3] to be suitable.

The application QoS parameters and network QoS are clearly different ways of talking about the behavior of a system – two different *languages*. We must provide a mechanism for communicating parameters in the appropriate language among the application end-point entities, as well as between the application and the network entities.

2.2 QoS Communication

A general architecture for communication of QoS parameters as requires two new services: **negotiation/renegotiation of QoS** and **translation of QoS**.

To characterize an actual negotiation, we ask *who the parties are?*, and *how do the parties negotiate?*. There are really two parties to any QoS negotiation in networked multimedia applications - other application elements and the network infrastructure. There is *peer-to-peer* negotiation between the application elements and *application-to-network* negotiation. This split between the types of negotiation is shown in Figure 2.

Peer-to-peer negotiation settles the multimedia requirements between the end-points. This process establishes an agreement between the parties with respect to the application QoS parameters. Using a separate control connection (non-parameterized) the application QoS are exchanged. The receiving party checks the incoming multimedia quality and service requirements for feasibility (e.g., resources, service existence, device support). The result is one of: "accept", "modify" , or "can't communicate". In the case of a "modify" answer, the receiver must modify its own sensory environment so that it can communicate with the sender. We assume that the sender sends its best possible quality of the media, and the receiver has to adjust if there are differences. This approach is a scalable approach especially for large-scale multicasting or broadcasting purposes.

The application-to-network negotiation communicates the performance requirements for the multimedia connections between the application and the network. This negotiation includes two steps. One step is **bidirectional translation of QoS** from/to the application QoS parameters to/from network QoS parameters and is implemented by the entity called the *QoS Broker*. The second step is **negotiation/ renegotiation of network QoS**.

Negotiation/renegotiation of network QoS establishes agreement between *connection management* and

[1] Figures 1-4 are at the end of the paper

network management on network QoS parameters. Negotiation/renegotiation of network QoS happens on a per-connection basis. The connections are unidirectional connections in our model. Thus, the connection set up is tied to negotiation of QoS parameters. The admission protocol performs actions to guarantee them (admission-reservation, admission-allocation) of resources at the end-points.

Detailed translation examples and the functionality of the QoS broker, as well as dependencies between negotiation/renegotiation process elements are further described in [7].

2.3 QoS Guarantees - Scheduling based on QoS

It is clear that the computer system software must participate in service delivery. The negotiation and admission control can offer their guarantees only if scheduling based on QoS is available as a service for QoS guarantees. The network QoS requirements have to be mapped further into a scheduler based on QoS.

An extensive body of work on scheduling based on QoS in switches exists; an example scheme is described by Hyman, *et al.* [12].

We partition traffic at the end-points into three basic classes using deadline times as the partitioning relation:

- hard real-time deadline media streams (tactile data & kinesthetic),
- soft real-time deadline media streams (audio, video streams),
- non-real time deadline media streams .

There are several reasons why time is the partitioning factor; among the more important are the fact that sensory data are continuous, and the observation that both computer operating systems and network resource allocation use time-division multiplexing (TDM) as a mechanism to implement their resource-sharing paradigms. Thus, one would expect mapping requirements to these resource providers to be more natural than in other traffic partitioning schemes.

For scheduling, we are using a mixed scheduling algorithm [16], which combines rate-monotonic and deadline-driven scheduling algorithms. An important architectural implication is that admission control must take into account the ability to schedule activities associated with the hard real-time streams.

3 Application to Telerobotics

3.1 Why Teleoperation

We are exploring our approach in the context of an actual application, that of telerobotics and distributed digital teleoperation [1]. The telerobotics/teleoperation application is a non-trivial application and provides many challenges to network research. The reason is that the involved media *sensory, data, audio* and *video*, with their complex application requirements, fill different portions of the QoS parameter space as it is shown in Figure 3.

ıTeleoperation allows a remote operator to exert forces or to impart motion to a slave manipulator. The operator can also experience the forces and resulting motion of the slave manipulator, known as "kinesthetic feedback". An operator is also provided with visual feedback, and possibly audio feedback as well.

Visual information requires at least megabit bandwidth with frame rates in excess of ten frames per second. Normally, teleoperation makes use of two to three video channels. The kinesthetic communications channel is required in both directions for each manipulator. There are normally two manipulators. Kinesthetic channels require transmission of a few hundreds of bits per sample, at about a 1000Hz rate. As one might expect, there are strict timing requirements on manipulator channels ("robotics data") - irregular or missing data can result in physical damage. Along with these channels might be channels for audio, and video information. This application has dynamic changes in its requirements over its execution because the physical information changes as the robot arms are mobile - they may obscure or expose a camera while moving. The changes of the physical information may result in renegotiation of requirements among the remote sites as well as changes in network guarantees. The complex timing requirements of the telerobotics application gives us a realistic and demanding platform with which we can study parameterized call/connection management and negotiation services. Telerobotics employs distributed control and execution mechanisms which force some real-time requirements on the network. As we experiment with lower layer protocols such as ATM Adaptation Layers (AALs), we expect to do considerable performance measurement and evaluation of the underlying ATM network from the application performance point of view.

3.2 System Configuration

Our test system configuration is shown in Figure 4. The network solution employs point-to-point links star-coupled to a high-speed ATM switch to form an ATM LAN. The LAN is interconnected to WAN facilities through the same switch; links are fibers operating with SONET OC-3c or OC-12. We operate ATM over SONET OC-3c, giving a bandwidth of 155 Mbps. ATM over SONET provides the key features of low error rates, high-bandwidth, and facilities which can be used for the paced data delivery we have discussed earlier in this paper.

The communication software and hardware support for video, audio and ATM host interface have been implemented on IBM RISC System/6000 workstations using AIX. To obtain robotics sensory data over the ATM network we connect the SUN-4 and RS/6000 stations with an S bus-to-Microchannel bus interconnection card at the slave side. On the master side a real-time processor (called "JIFFE" and labeled as such in Figure 4) and a dedicated IBM PC provide the robot control. A BIT3 bus connector card connects the JIFFE processor with the IBM RS/6000 workstation. The hardware and device drivers on the RS/6000 are functional; we have achieved application-to-application bandwidths up to 130 Mbps[17] and feedback loops operating successfully at over 500Hz. This 500Hz figure is achieved at the interfaces between the robotics control/communication unit; and should be sustainable over small ATM networks. While the interarrival time for a 500Hz loop can be maintained over an ATM WAN (we intend to study this in the AURORA network), we realize that better modeling in the robotics controller subsystems will be necessary to cope with the increased feedback delays due to longer distances.

The specification and design of the QoS broker as well as negotiation/ renegotiation of application QoS are implemented as part of the telerobotics project.

Currently we are working as on the implementation of network guaranteed services for our ATM network using ideas described in [3] and [4], as well as the extended connection and call management, (including negotiation/renegotiation of network QoS) described in section 2.

4 Conclusion

The main contribution of this paper is an architecture for propagating an integrated sensory environment between elements of a complex distributed application. The approach is based on two basic constructs in the network.

The first construct is multiple points of service negotiation and renegotiation, and system support for these services. We discussed both peer-to-peer and applications-to-network (layer-to-layer) points of service, and showed how a *language translation* model helps to solve this problem.

The second is the mechanism necessary to deliver the negotiated services. We indicated that the mechanism must include translation of QoS parameters into activities required by other entities, such as operating system schedulers, logical multiplexing and demultiplexing in protocol design, and network call admission strategies.

We are using teleoperation of intelligent machines as a driving application. The experiments are underway, and we hope to understand the issues in such a way that we can prioritize our concerns, rather than simply enumerating them as we do today. Only such experiments can provide the necessary understanding. Important open questions include the nature of optimizations possible in combining multimedia data within an application, use of sophisticated planning and remote device status models to reduce network traffic, and the traffic implications of the dynamics (e.g., the magnitude and duration of traffic bursts) inherent in this class application.

References

[1] Ruzena Bajcsy, David J. Farber, Richard P. Paul, Jonathan M. Smith, "Gigabit Telerobotics: Applying Advanced Information Infrastructure", *Technical Report MS-CIS-93-11*, January 1993

[2] Jae-il Jung, Dominique Seret, "Translation of QoS Parameters into ATM Performance Parameters in B-ISDN", *IEEE INFOCOM '93*, San Francisco, CA, March 1993.

[3] Jean Ramaekers and Giorgio Ventre. "Quality of Service Negotiation in a Real-Time Communication Network", *Technical Report TR-92-023*, International Computer Science Institute, Berkeley, CA, April 1992

[4] Dinesh C. Verma "Quality of Service in ATM Networks", *Technical Report* University of California, Berkeley, CA, 1990

[5] Hui Zhang, Srinivasan Keshav. "Comparison of Rate-Based Service Disciplines", *SIGCOMM'91*

Conference, Communications Architectures and Protocols, September 1991, Zurich, Switzerland, pp. 113-121

[6] David D. Clark, Scott Shenker, Lixia Zhang, "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism" *SIGCOMM'92, Conference on Communications Architectures & Protocols*, Baltimore, Maryland, August 1992, pp.14-26

[7] Klara Nahrstedt, Jonathan M. Smith, "Revision of QoS Guarantees at the Application/Network Interface", *Technical Report No. MS-CIS-93-34, University of Pennsylvania*, March 1993.

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

[9] 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.

[10] 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.

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

[12] Jay Hyman, Aurel A. Lazar, Giovanni Pacifici, "Joint Scheduling and Admission Control for ATS-based Switching Nodes", *SIGCOMM'92*, Baltimore, Maryland, August 1992

[13] 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*, Vol. 25(6), January 1993.

[14] 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.

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

[16] C. L. Liu, James W. Layland, " Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment", *Journal of the Association for Computing Machinery*,Vol.20, No.1 January, 1973, pp. 46-61.

[17] C. Brendan S. Traw and Jonathan M. Smith "Hardware/Software Organization of a High-Performance ATM Host Interface", *IEEE Journal on Selected Areas in Communications (Special Issue on High Speed Computer/Network Interfaces)* Vol. 11, No. 2, Feb. 1993, pp. 240-253