System Support for Multimedia Applications

Joseph Pasquale, George Polyzos

Computer Systems Laboratory
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093

September 1990

Abstract

We describe work on system support for multimedia applications currently underway at UCSD. We are focusing on operating system I/O mechanisms and network multicasting algorithms. The project is comprised of two assistant professors (the authors), seven graduate students, and one undergraduate.

1. Introduction

We have recently started a project to investigate issues of system software support for multimedia applications, especially those which have digital audio and video I/O components. In the area of operating systems, we are investigating I/O device abstractions, and mechanisms which support high-bandwidth low-delay kernel data paths between devices. In the area of networking, we are investigating efficient multicasting under real-time constraints. We now briefly describe each of these topics.

2. Operating System I/O

With the growth in number and variety of I/O devices, particularly those supporting multimedia, we see the role of the operating system of a workstation tending more and more toward that of an I/O scheduler, router, and switcher. To achieve a flexible programming environment, we are developing an I/O device model to serve as a single uniform abstraction for process-device and device-device interactions. The requirements are that the model must be rich enough to capture those aspects of real-time high-bandwidth devices which allow for efficient programming and operation, yet simple enough so that processes may interact with dynamically bound devices conveniently and uniformly.

Our search for a uniform I/O abstraction is similar in spirit to Cheriton's UIO interface [1]. However, we focus on the inclusion of devices which make up a distributed multimedia computing environment: real-time capture or presentation devices (e.g., microphones, cameras, frame buffers), as well as more traditional devices such as disks and network interfaces. Our model allows for the treatment of physical devices, such as those just described, as well as virtual devices such as windows (i.e., virtual terminals) or stream network connections, in a uniform manner. For instance, a process is able to read from a camera and write to a window, treating them as devices which, despite having different attributes, have the same interface.

The design of a uniform interface must be accompanied by the determination of underlying kernel mechanisms necessary to support fast and high capacity I/O, as dictated by model specifications as well as the interface parameters such as buffer size, flow control, or more general "quality of service" parameters, such

This work is being supported in part by a Presidential Young Investigator Award by the National Science Foundation, Digital Equipment Corporation, NCR Corporation, and the Powell Foundation.

as those provided in DASH [2]. An example of such mechanisms we are considering are buffering and processing modules, similar to Ritchie's stream modules [3], but which can be scheduled independently or as logical groups, with support for continuous (in addition to discrete) media transmission. One area of particular interest is the composition of these I/O modules to form in-kernel data paths directly connecting continuous-media devices, such as a camera and a video window. Finally, the composition of modules is not limited to linear formations, as we expect group communication to be common in distributed multimedia applications, such as conferencing.

Such multiway multimedia connection mechanisms are an important area of investigation; Leung [4] has found connector objects very useful in multimedia applications. We see the network as having a significant effect on the design of connection mechanisms, especially when multiple distributed process are being connected for group communication. Thus, we are developing an abstraction for group communication over a network, which conforms to our uniform I/O model, which we call the "multicast channel." Briefly, a process may allocate a free multicast channel from a large pool accessible by any process in the distributed system. Thereafter, processes may join the multicast channel; this allows them to send data frames into the channel, with all processes on the channel receiving them. Data frames are typed, timestamped, and prioritized, so that soft real-time guarantees on a per-type basis with congestion control can be implemented. Whenever a process joins or leaves, all processes which are on the channel receive interrupts (e.g., Unix signals) so that they may take appropriate actions.

This multicast channel is particularly useful for multimedia group applications such as conferencing, as it allows processes to dynamically direct the transmission from various continuous-media input devices (e.g., camera and microphone) to multiple remote windows (e.g., video and audio [5]) in a synchronized one-to-many fashion. When a user joins the conference, thereby causing a process to join a multicast channel, interrupts generated at other processes cause them to create windows for the new user (the opposite occurs when a user leaves). Its implementation relies on underlying network support which constructs efficient multicast trees, which we discuss next.

3. Network Multicasting Algorithms

Efficient multicasting becomes increasingly important in the context of packetized audio and video, because of the very high bandwidth demands of these media, even in the presence of high-speed networks specifically designed for digital transmission of continuous media (e.g., Broadband ISDN, etc.). For example, in the recent experiments carried out by ISI and BBN [6], using a Double-Queue, Dual-Bus type of (linear) network architecture to transmit packet video, efficient multicasting was used by avoiding duplication of the video packets until the last possible point (when the packets had to ''leave'' the dual bus).

There are already many interesting results in the area of efficient multicasting [7], but optimal solutions, under most circumstances, are not available. The problem is known in the graph theoretic literature as the (minimum) Steiner Tree problem:

> Given an undirected distance graph G(N,L,c), where N is the set of nodes (vertices), L is the set of (bidirectional) links (edges), and c a cost function from L to the non-negative integers, and a set of nodes of interest, i.e., the multicast group, M (M a subset of N), find a tree in G, spanning M, with minimal cost.

Unfortunately, it has been shown that the Steiner Tree problem is NP-complete [8]. On the other hand, a fast algorithm ($O(|M||N|^2)$), due to Kou, Markowsky, and Berman [9], gives solutions with cost no more than twice the optimum, and usually much better.

However, when one considers continuous media, additional constraints are introduced because of the real-time requirements usually associated with these media, as for example in the case of multimedia conferencing. Not only do we want to minimize some cost function associated with the communication (typically, transmission bandwidth), but at the same time we want to make sure that the solution remains feasible from a delay point of view. Notice that minimizing bandwidth usage and delay experienced are many times conflicting goals. In particular, delay considerations such as propagation delay (which can be significant for satellite channels and coast-to-coast links), transmission time (if links are not very fast), processing time (if the number of links in tandem is large), and also variations in delivery time (which can increase considerably with the number of intermediate nodes), can make an otherwise perfect solution infeasible for continuous media. On the other hand, some of the difficulties traditionally associated with multicast (and broadcast), for example atomic or reliable delivery to all members of the multicast group, might be less important in the context of continuous media where error-free transmission requirements are usually relaxed in order to achieve better (e.g., real-time), delay characteristics.

We are investigating the Steiner Tree problem under additional (delay) constraints. For example, let us introduce an additional weight function on the links, d > 0, and distinguish one of the multicast nodes, S (S in M), as the source. If then we introduce the additional requirement that the d-cost between the source and each member of the multicast group must be constrained, let us say by D (for simplicity we assume a uniform delay requirement), we have a realistic model of the output problem of a participant in a multimedia conference. We refer to this problem as the Rooted Constrained Steiner Tree problem (RCST). Note that this problem is not trivial even if all the c and d weights are equal to one.

Alternatively, if no source is distinguished, and the constraint is modified so that the sum of the d-metrics along the path between any two members of the multicast group is constrained by a given quantity, we obtain a common transport tree that could serve as both input and output for all members of the group (assuming the required capacity is available). A solution to the RCST problem constitutes the basis for the establishment of a multicast channel, as discussed above. Thus, questions of distributed implementation of solutions, as well as dynamic reconfiguration, arise naturally. Note, that if the d-metric is a delay guarantee, as described by Ferrari [10], offered, for example, by the node before the link in question, this procedure would identify a minimum cost guaranteed performance multicast channel.

Preliminary results show that heuristics based on (i) the branch-and-bound technique, (ii) an Essau-Williams type of branch exchange [11], and (iii) an extension of the Kou-Markowsky-Berman (KMB) heuristic [9], are very promising (particularly for graphs similar to realistic network topologies, where the degree is relatively low). To evaluate their effectiveness, we are performing comparisons of the obtained heuristic solutions with (i) the optimal, (ii) the unconstrained KMB heuristic, (iii) the MST based on the d-metric, and (iv) random trees (satisfying the delay constraint), on a number of random graphs and actual (or historical) network topologies.

4. Project Status

As we are in the exploratory stages of our research, we have chosen an experimental approach. In conjunction with the development of various abstractions and algorithms discussed above, we have been involved in the building of various distributed real-time multimedia applications, such as a simple audio conferencing program, using standard system software (e.g., SunOS 4.0, X windows, TCP streams) to discover where problems arise in existing system software, and consequently, whether new or improved interfaces and/or mechanisms are needed. Of course, many of the problems we have encountered, such as lack of real-time control over processes, were predicted from the start, but we felt we needed more of a quantitative understanding of the problems.

The project began in April 1990. The project members include two assistant professors, J. Pasquale and G. Polyzos, seven graduate students, Eric Anderson, Mark Bubien, Kevin Fall, Vach Kompella, Keith Muller, Dipti Ranganathan, Robert Terek, and one undergraduate, Scott McMullan.

5. References

1. D. Cheriton, "UIO: A Uniform I/O Interface for Distributed Systems," ACM Trans. Comput. Sys., Vol. 4, No. 1, pp. 12-46.

2. D. Anderson, "A Software Architecture for Network Communication," Proc. Eighth Int. Conf. Distributed Comput. Syst., pp. 376-383, IEEE, June 1988.

3. D. M. Ritchie, "A Stream Input-Output System," ATT Bell Laboratories Tech. Journal, Vol. 63, No. 8, pp. 1897-1910, October 1984.

4. W.H. Leung, L.F. Morgan, M.J. Morgan, and B.F. Wong, "The Connector and Active Devices Mechanisms for Constructing Multimedia Applications," Second Workshop on Workstation Operating Systems, pp. 68-72, IEEE, September 1990.

5. L. F. Ludwig, N. Pincever, and M. Cohen, "Extending the Notion of a Window System to Audio," IEEE Computer, Vol. 23, No. 8, pp. 66-72, August 1990.

6. S. Casner, K. Seo, W. Edmond, and C. Topolcic, "N-Way Conferencing with Packet Video," Third International Workshop on Packet Video, Morristown, NJ, March 22-23, 1990.

7. M. Ahamad, ed., Multicast Communication in Distributed Systems, IEEE Computer Society Press Technology Series, Los Alamitos, CA, 1990.

8. R. M. Karp, "Reducibility Among Combinatorial Problems," in Complexity of Computer Computations, ed. R. E. Miller and J. W. Thatcher, pp. 85-104, Plenum Press, New York, NY, 1972.

9. L. Kou, G. Markowsky, and L. Berman, "A Fast Algorithm for Steiner Trees," Acta Informatica, vol. 15, pp. 141-145, 1981.

10. D. Ferrari, "Guaranteeing Performance for Real-time Communication in Wide-area Networks," Tech. Report TR-89-001, International Computer Science Institute, Berkeley, CA 94720, January 1989.

11. L. R. Essau and K. C. Williams, "On Teleprocessing System Design," IBM.