The presentations in this session spanned a wide range of levels of abstraction, from novel low-level subsystem designs, through entirely new operating systems, and up to high-level distributed system architectures. However, the presentations all touched on various aspects of how operating systems can be made to provide enhanced support for digital audio and video. There are a great many ways in which system software can be made to actively support the unique requirements posed by applications which include continuous digital media. One should not expect operating systems to simply “get out of the way” and permit applications direct access to the underlying hardware. There are still a number of valuable things that can be done at the system level to meet the needs of multimedia applications.

The session’s first full paper was presented by Valerie Gay of the University of Paris and described how industry standard distributed object system architectures can be augmented to permit the specification of interfaces for objects that possess timeliness constraints. In particular, the suggestion was made to modify OMG-IDL to include a means of specifying streaming constructs, as well as including the notion of “computation environment contracts” which provide the additional information needed to capture all of the (higher-level) quality of services requirements of objects. The speaker made a case that it is becoming increasingly important to incorporate the means to express the needs of real-time multimedia applications into the formalisms and tools for architectural specification of development of large-scale, object-based, distributed systems.

The second paper of the session was presented by Mike Jones of Microsoft Research, who described a novel approach to real-time system resource management being implemented in the Rialto operating system. This work illustrated how an operating system can be devised which provides a new set of abstractions that reflect the needs of time-critical (e.g., multimedia) applications. Specifically, the Rialto OS provides a means for effectively resolving the forms of contention which occur when independently-developed real-time applications compete for shared system resources. The presentation suggested that abstractions provided by Rialto, in conjunction with the (local) resource planning modules, provide a simple, yet effective means by which successful resolutions to difficult resource allocation problems can be achieved.

Kevin Jeffay of the University of North Carolina presented the final full paper which described an approach to real-time computing that defines application timeliness requirements in terms of execution rates – which is translated to the allocation of a given amount of computational resources, over a given interval of time. Included in this presentation was a description of some initial results of the application of this rate-based technique to a desktop teleconferencing application. The benefits of the proposed computing model included a straightforward method of expressing the timeliness behavior of applications, as well as a means for applications to dynamically adapt to changes in system resource availability.
In the first of the adjunct paper presentations, Chris Lindblad made some suggestions as to how operating systems could better provide support for multimedia applications. These suggestions were made from the perspective of the designer and implementer of a substantial, application-level, multimedia system (i.e., the VuSystem at MIT). The suggestions for operating systems support included a desire for additional resource availability feedback to the applications from the system, as well as for finer-grained processing control for applications.

Next, Guru Parulkar of Washington University presented a proposal for a new programming interface for I/O subsystems that provides support for multimedia applications in the form of reduced system overhead and improved application performance. The proposed operating system enhancements focussed on the creation of mechanisms to reduce (or better manage) the number of system calls, interrupts, and instances of memory-to-memory copy operations involved in each instance of I/O.

Finally, Hiroshi Kitamura of NEC discussed his interest in high-performance network interfaces, his analysis of TCP performance that led to the observation that memory copying incurs a significant cost, and his experience with an implementation of a zero-copy TCP interface on an OC-3 ATM network. The results presented supported the view that zero-copy TCP implementations can provide performance gains (where packet checksums need not be computed in software).

Broadly speaking, the round-table discussions centered on issues related to interactions between the application and the operating system, the problem of maintaining a connection between user-level views of quality of service and system-level resource allocation decisions, and the nature of user expectations and their impact on system software.