Multimedia Services for Distant Work and Education in an IP/ATM Environment

David Fernández and Ana B. García
Technical University of Madrid

David Larrabeiti and Arturo Azcorra
Carlos III Univeristy of Madrid

Piotr Pacyna and Zdzislaw Papir
AGH University of Technology

We developed, tested, and evaluated multimedia distant education applications running over an IPv6/ATM-based broadband access network. We improved the quality of service and adapted a set of distance education applications—including Digital Video Library, Virtual Workspace, and video–audio conferencing tools—to work over IPv6 and let users control the QoS.

Many universities and research sites have years of experience in applying telecommunications to education. More recently, multimedia technology has created several applications to support the educational process.

Although within an internal network it's possible to find advanced high-quality tele-education services, the most common communication infrastructure for exterior sites is based on Internet and/or Integrated Services Digital Network (ISDN) in a point-to-point setup for video conferencing. The Internet provides no quality of service (QoS) guarantee and a low bandwidth. ISDN is expensive, difficult to integrate with local area network (LAN) communication infrastructures, and doesn't provide network multicast. The fact that only the Internet and ISDN are available for service provision inhibits the development and deployment of advanced tele-education environments based on the already available and cost-effective technology. Advanced virtual reality and virtual presence applications already exist, but they require advanced communications infrastructures.

Standardization bodies, telecommunications operators, content providers, and the information technology industry are working to create the basis for new applications. Such applications include remote surgery, distance learning, video and audio on demand, videoconferencing to the desktop, videophone, teleshopping, remote process control, or positioning systems. These new applications must control the QoS through bandwidth allocation, priority of flows, security, mobility, and some may require precise timing and synchronous service. Most importantly, however, they require connectivity.

Hence, the infrastructure must support the applications with a predictable and user controllable performance. Currently, no global technology provides that to a satisfying degree. Emerging technologies in this area still suffer from lack of sufficient standardization, coverage, capacity, or acceptance.

Asynchronous transfer mode (ATM) provides a good basis for controlling quality, resources, and billing. On the other hand, ATM lacks standard applications, interoperability, international connectivity, and global deployment. For a seamless communication, the ATM networks so far don't support dynamically established connections. The Internet is independent of the underlying subnetworks, has a high level of connectivity, and an enormous drive in applications' developments. The next generation of the Internet protocol (IPv6) solved the problem of the scarcity of addresses, while at the same time incorporating some improvements to support advanced applications. The Internet has some drawbacks, as it doesn't provide a secure and reliable transmission, and it can't support synchronization and jitter control. Furthermore, it uses a packet store and a forward of full packets that implies those transmission delays will usually be higher than for ATM.

Given the pros and cons of ATM and Internet protocol, it's apparent that none of the protocols will solve all the needs. But combining these two families, the requirements of most applications can be met to a satisfying degree.[1,2]

Emerging streaming media applications in the Internet primarily use the user datagram protocol

(UDP) while best-effort data transmission-oriented services prefer the transmission-control protocol (TCP/IP) suite. The most recent research[3] reveals that UDP-based streaming can cause a congestion collapse and an unfair allocation of bandwidth among competing traffic flows. The suggested remedy is the ATM available bit rate (ABR) transport in the first-mile, last-mile, and core networks for both types of Internet applications.

To fulfill the general demand for improved quality of Internet services, we developed and demonstrated a concept for improved QoS based on the integration of IPv6 and ATM through our European Union-funded Broadband Trial Integration (BTI) project. The project focused on implementing QoS in an ATM-based passive optical network (APON),[1] supporting unicast and multicast with a well-defined QoS control.

To explore QoS features offered by the integrated IPv6/ATM networking environment, we specified a set of distant education applications, including the Digital Video Library (DVL), Virtual Workspace, and video–audio conferencing tools. The Virtual Workspace is a set of integrated data-conferencing applications that support collaborative education. It's basically reused from the Leverage EU-funded project,[4] and it lets users create virtual meetings and interact in real time by sharing documents. The DVL consists of a video-streaming engine, database, and content manager server. The software embeds user interfaces for video retrieval, playback, uploading, and QoS control in a standard Web browser. The video–audio conferencing tool lets users establish point-to-point and multipoint videoconferences. It's based on the well known Videoconference (VIC) and Robust Audio Tool (RAT) multicast tools used in Internet Mbone. All these applications work over IPv6 and include a simple interface to let users control the QoS, basically by selecting the desired quality level.

We based the QoS-controlled network on IPv6, Resource Reservation protocol (RSVP), protocol Independent Multicast (PIM), Next Hop Routing protocol (NHRP) and ATM, with point-to-point and point-to-multipoint switched virtual connections (SVCs). Therefore, to support all these network technologies and fulfill QoS requirements of applications, we integrated a complete protocol stack running over end-user Windows NT stations from the existing public protocol blocks available on the network (mainly RSVP and IPv6 implementations) with missing IPv6 over ATM drivers added.

We measured the network's technical performance to evaluate the proposed concept's viability. In addition, we performed a program of structured usability testing to evaluate the user perception of the QoS control and the user interface. For this purpose, we connected students and teachers at universities and schools in Denmark, Poland, and Portugal to a trial network. We used several methods for the evaluation—an expert evaluation, think-aloud sessions, and focus group discussions. Through these methods we discerned the application's usability and the users' understanding of the network features.

Virtual Workspace

We based the Virtual Workspace used in the BTI project on the collaborative education environment developed in the Leverage project.[4] The environment consists of several integrated data-conferencing applications that let users create virtual meetings and interact in real time by sharing documents. We adapted the environment and all its applications to include user-controlled QoS support.

The following modules make up the Virtual Workspace:

- *Session desktop*—the entry point to the system. It lets users know which sessions (virtual meetings) are running in the system; create new sessions or join in the running ones, start new applications or invite new users to join during a session, and so forth. In general, the session desktop coordinates the applications running in a session.

- *Chat tool*—a simple multipoint text-conferencing tool, which lets users interact by means of short text messages.

- *Shared text editor*—lets users share text files. It's a simple text editor—controlled by a single user at a time—that shows exactly the same view of the text in every participant's window. The user controlling the editor can create a new text document, load or save a text from or to a disk file, modify the existing document, and so on. Other participants can ask for the right to control the editor.

- *Shared Web browser*—a synchronized Web browser that displays the same Web document in every participant's window. As in the shared text editor, only one user at a time can control the application, although the right to control

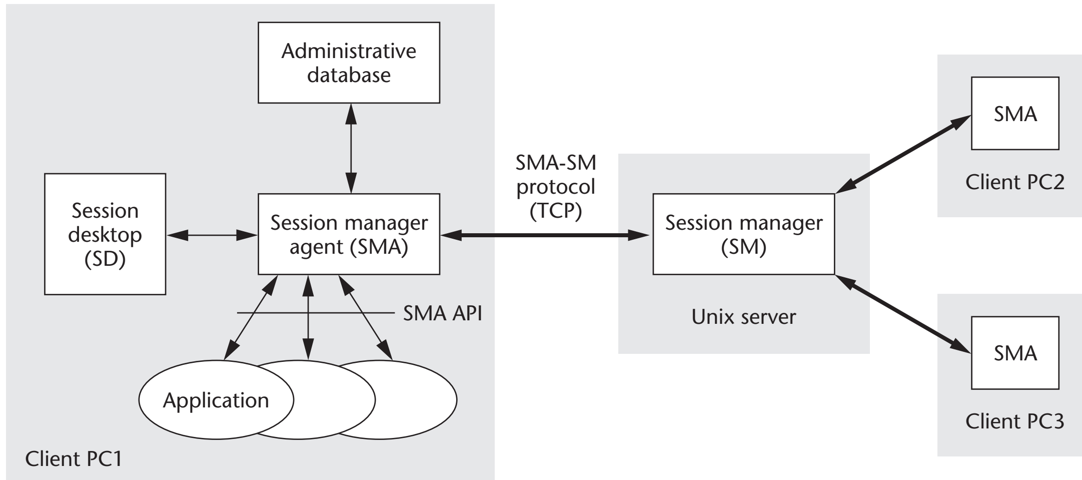

Figure 1. The Virtual Workspace architecture.

it can be passed on between users. It includes telepointer functionality to help users mark documents during navigation.

- *Shared blackboard*—a shared drawing application where every participant can draw at the same time using typical blackboard tools like pen, ink, lines, circles, and so on. Pictures can be saved or retrieved from the disk.

Although BTI Virtual Workspace doesn't include audio or video tools, it's supposed to be used simultaneously with a videoconference or an audioconference application.

Application architecture

Figure 1 shows our Virtual Workspace architecture, which we based on a client–server model where client machines (Windows-based) connect to servers (Unix-based) to start or join in cooperative sessions.

Most of the Virtual Workspace modules run over client machines: the shared applications previously mentioned; the session manager agent (SMA) in charge of a session; the session desktop (SD), which is the user interface to the system; and other auxiliary modules to access the administrative database that stores information about registered users. Servers only run the session manager (SM), a module responsible for the global coordination of sessions. The session manager can be distributed between several physical machines, allowing multisite cooperative sessions. For example, we used this configuration in the Leverage project final trials, with three sites in Madrid, Spain; Cambridge, UK; and Evry, France.

Application QoS model

We adapted the Virtual Workspace to the QoS-enabled network with several initial assumptions.

Characterizing the traffic generated by the different tools usually leads to an unpredictable traffic pattern during the session that heavily depends on the users' behavior. However, since the tools included in Virtual Workspace are all interactive, they're typically used sequentially in a conversational way. Therefore, we decided that instead of making network resource reservations for each tool, it was reasonable to make a shared reservation for all the traffic generated by the tools used in a cooperative session. To simplify the management of the shared reservation, we reprogrammed all the tools to use a common data channel for client–server communication.

This way, a single TCP connection carries all the data of the cooperative session and hence, it needs only two simple unicast RSVP reservations per client (a reservation for server to client traffic and another for client to server traffic). Such a design reduces the amount of ATM network layer connections to be set up, yields less management complexity, and enables the integrated control of the traffic produced by the whole set of tools. This issue is more important than you would expect—it might compromise the system's viability, since scalability is a well-known weak point of the integrated services QoS model.

Furthermore, loosely coupled tools greatly complicate the adaptation of Virtual Workspace, since QoS awareness should be included in each application tool. Once we modified the communication model, we added RSVP support to the system—still running over IPv4—and enhanced the user interface to include the QoS control, as we show in

Figure 2. Basically, users can choose between several levels of service, from no QoS (best-effort behavior) to high quality. The applications internally translate these service levels into several increasing RSVP-controlled load reservations that represent different responsiveness grades.

Finally, we migrated the communication modules to use IPv6. In general, the migration to IPv6 wasn't difficult from the programming point of view, mainly due to the similarity of network Application programming interfaces used in IPv4 and IPv6 protocols. The most important hindrance was the immaturity of the protocol stack prototypes available at the time we carried out the migration.

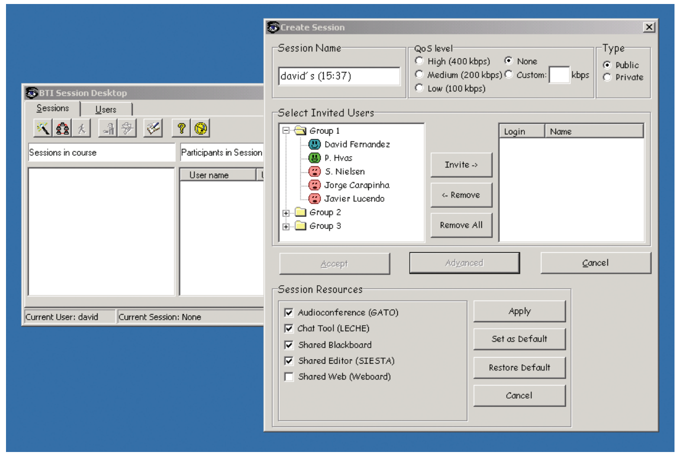

Figure 2. Creating a cooperative session using Session Desktop interface.

Digital Video Library

Video drives distance learning applications. It's gradually infiltrating into the workspaces of many research institutions and universities, either in the form of videoconferencing, multimedia electronic mail, remote access to educational and training recordings, or remote lecturing.

The DVL is a network application that offers a video-on-demand service. It lets users retrieve videos from a server over the network, controlling its reproduction through common VCR functions. Groups share video content efficiently by using multicast network facilities. It also includes a content database and associated tools to make queries for specific contents.

Apart from its applications in distance learning, the DVL can be used in e-market niches like e-commerce, real estate, telemedicine, and entertainment. The whole setup could also be deployed as a demonstration kit for showing broadband networking equipment capabilities.

Today there are many commercially available video streaming engines like RealSystem Server from Real Networks and QuickTime Streaming Server from Apple Computer.[5] In addition, University of Maryland, the Massachusetts Institute of Technology, and Purdue University[5] have developed other platforms for video retrieval.

The DVL is a complete platform for an Internet service provider (ISP) focused on querying, retrieving, and streaming video content to customers. DVL's main features include a flexible user interface based on the HTTP browser, both unicast and multicast addressing styles, and a user-oriented control of QoS requirements in IP networks.[5]

Application architecture

The main design requirement for the DVL was to build it from existing applications and adapt them to explore the QoS features that the integrated IPv6/ATM transport offers.

We made the server from a video-streaming engine based on the Oracle Video Server (OVS) product combined with a video material storage, a content manager database, and a Web server (Apache) with several auxiliary scripts that let Web-based user interfaces control the whole application. Clients are PC based and include a standard Web browser (Microsoft Internet Explorer) augmented with an OVS player plug-in, as well as auxiliary OVS Active X controls for VCR-like functions (start, stop, rewind, forward) and dynamic objects (scrollbars and counters) that manage a video-stream reproduction. Figure 3 (next page) shows the DVL's architecture while Figure 4 shows the user interface.[6,7]

Yet another design criterion was to equip the DVL with content managing capabilities that support remote users when searching and querying for a specific video record.[7] The content database stores data and textual descriptions of all available video recordings with the technical data necessary for their correct reproduction. Technical data include the name and the size of the associated Moving Pictures Expert Group (MPEG) file, the display dimensions, the encoding technique with the bit rate used during the video's production, the film length, and resource reservation parameters (passed to a transport network).

Additionally, the catalog data lets users query for particular content, including categories and subcategories, film titles or their episodes, and associated keywords. We indexed all the DVL content to facilitate the queries through the content manager interface. When users access the content manager server, Common Gateway Interface (CGI) scripts usually result in System Query Language (SQL) queries to the Postgres-based database and prepare HTML pages to show the results (Postgres is an open-source database management system).

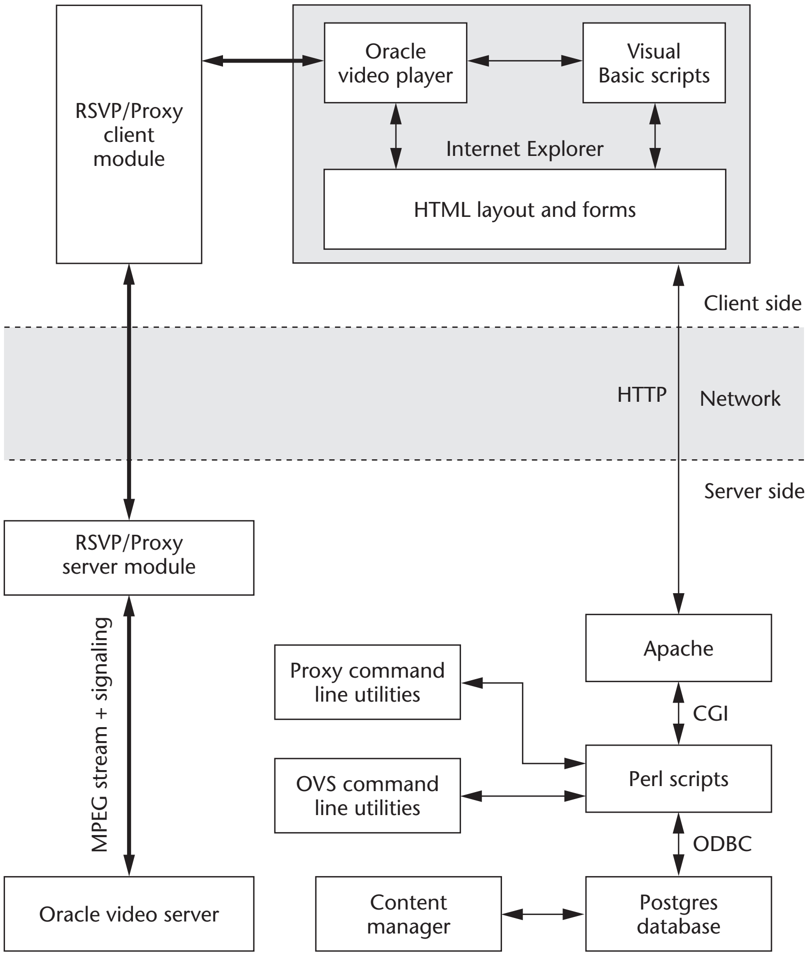

Figure 3. DVL's architecture.

Figure 4. User interface—a play page.

The DVL user interfaces are completely Web based. The Web server processes users' requests made through the client browser, either through HTML pages or by running CGI scripts written in Perl. The server even performs authorization functions based on standard username–password schemes. Visual Basic scripts running from inside the Web browser handle events on the user's side and check whether data filled in the HTML forms are correct. Using Web-based interfaces minimizes the software installation effort needed in client machines and reduces the amount of time to learn how to use the application.

Halsall[8] discusses general rules for developing digital video libraries suited for the IP networking environment in more detail.

Application QoS model

The QoS that users perceive involves several factors: response time, resolution (dimensions) of the displayed video, color fidelity, and stability of the playback process (no freezes, glitches, or discontinuities). To play back a video through the network with a satisfactory quality, we must map all these parameters to the transport network's QoS parameters.

Anderson et al.[9] and Gozdecki et al.[6] present more thorough discussions of mapping rules. For our purposes, we assumed that the ATM constant bit rate (CBR) service will be the most appropriate transport for the video content. We based our assumption on the output buffering implemented in the OVS streamer and ATM CBR connections available between Poland, Portugal, and Denmark during the project lifecycle.

We estimated that the requested bandwidth was about 1.2 megabits per second (Mbps) for low or medium picture quality or between 2.5 to 4.0 Mbps for good quality. Output buffering in the OVS engine greatly affects typical QoS parameters like application response time, delay jitter, and packet loss, but the parameters weren't subject to any serious tuning.

Adapting the DVL to work over IPv6 with QoS support included using two proxy software modules running on either side of a client–server configuration that concentrated all the functions pertinent to IPv6, multicast, and QoS support. As we show in Figure 5, these two modules intermediate all the communications between

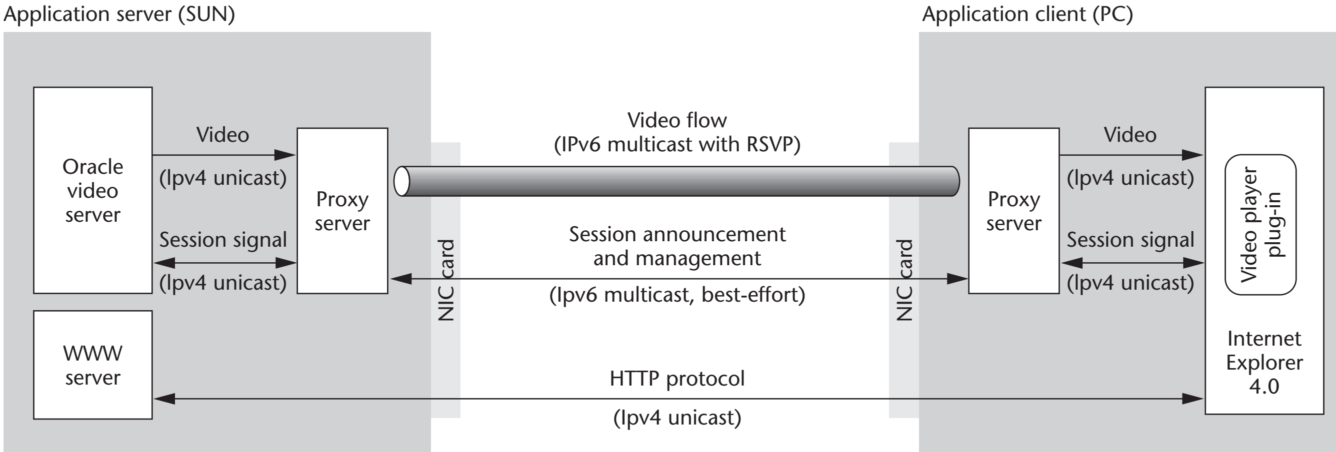

Figure 5. QoS architecture in the DVL.

the OVS engine and the video player plug-in.

Whenever a user makes a choice through the content manager interface and decides to retrieve a video from the server, this launches both proxy modules. The client sends a request to OVS to reproduce the video, but instead of giving it the client machine's address, it provides the proxy server address, directing all session control information and the video stream to the proxy server module.

The proxy server module converts all traffic from IPv4 to IPv6 (and in case a multicast session has been requested, from unicast to multicast) and forwards it to the proxy client module, which makes the opposite conversion before giving the information to the video player plug-in.

The adopted QoS approach follows the integrated services model defined by the Internet Engineering Task Force (IETF) with clients using the RSVP protocol initiating the resource reservation setup.[10] In the DVL, proxy modules establish the RSVP reservation for video flows. After processing a user's query for a particular recording, the server retrieves corresponding reservation parameters from the database and passes them to the proxy server module. The server module requests its local RSVP daemon to create a new session directed to the proxy client module.

The resource reservation applies only to video traffic. The control traffic interchanged between proxy modules and hypertext transfer protocol (HTTP) traffic between Web browser and Web server is sent through best-effort channels.

The system administrator assigns default values to RSVP parameters for each video recording when upload it to the to the server mass storage. RSVP parameters are derived from the prior MPEG encoding process, which sets them for a high-quality playback. This approach ensures that inexperienced users receive the video content at a satisfactory QoS level.

Alternatively, experienced users may select a manual operation mode, which gives them the option to change the RSVP defaults and trace the RSVP reservation progress. Providing users with a choice of different encoding settings (high, medium, and low quality) can complement this approach.

Adapting of services to networking technologies

For users to access network services from applications, we needed an integrated protocol stack that would work with Windows NT. The protocol stack should provide IPv6 over ATM with full dynamic multicast support via the multipoint SVC mesh approach. It should also provide RSVP over IPv6/ATM, following the integrated services model and offer a standard API to the applications to access QoS-enhanced network services.

Because no complete solution for the Windows NT platform was available, we developed the protocol stack Ipv6 over ATM Adaptation Module (PATAM)[11] from existing public protocol blocks available on the Internet. Except for implementing all the complex functions needed to run IPv6 over ATM, including multicast address resolution server (MARS) client, the protocol stack maps the QoS requirements specified by applications to the ATM network resources. The stack dynamically establishes all the necessary point-to-point and multipoint SVC ATM connections, either with the router or with other participants, using unspecified bit rate (UBR) service for best-effort traffic or CBR for QoS traffic. In addition the protocol stack has to classify and schedule the transmission of IPv6 packets through the connections created.

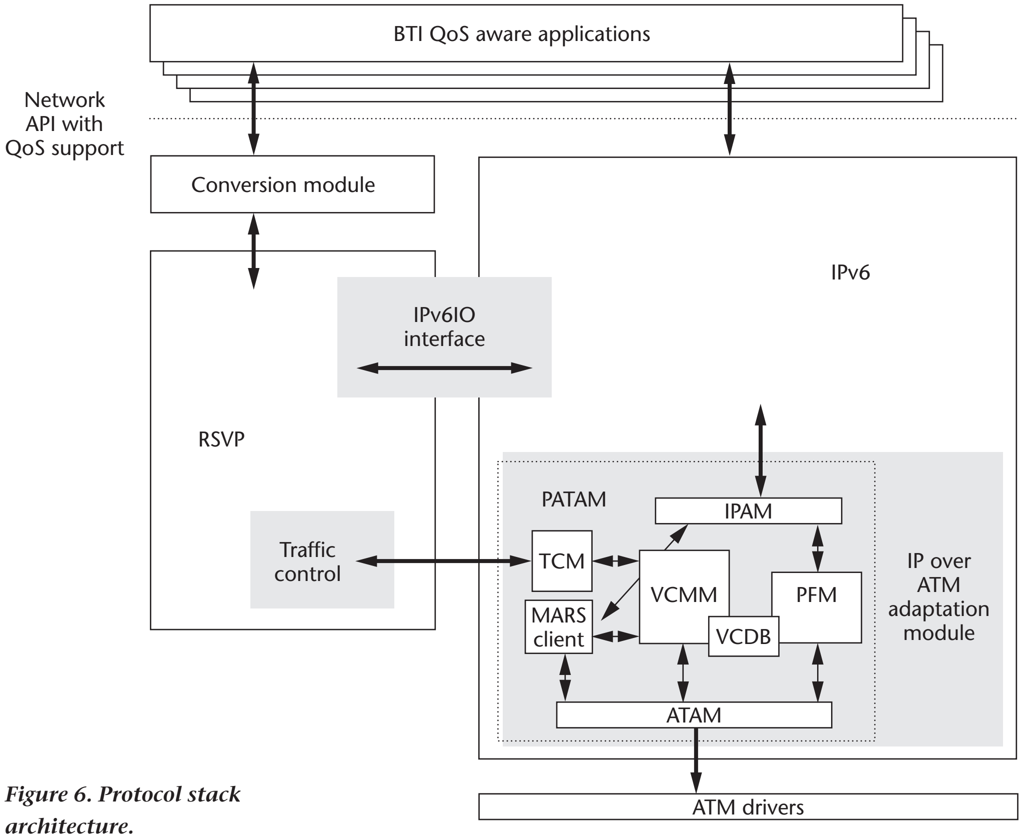

Figure 6. Protocol stack architecture.

and receives data, and the other controls the reservations requested for data flows.

Evaluation

As mentioned before, the BTI project aimed to demonstrate a way to provide QoS support in distant work and education applications in a broadband access network. We put a lot of effort into developing and integrating the variety of involved technologies. We also evaluated the system, including the technical and usability issues.

From the technical side, we verified the behavior of the applications with qualitative and quantitative measurements regarding the service performance for different QoS levels under various network loads. As we expected, QoS support provided the applications with the guaranteed network resources they needed to work even under a heavy load. We also conducted other tests to evaluate, for example, the efficiency of network protocol mapping (IP/ATM) or the resources used in typical scenarios, aiming to evaluate the solution's scalability.

Basically, the application and network developers performed the technical evaluation without the participation of endusers. However, since the network performance with respect to QoS substantially impacts the subjective quality of applications, we conducted final performance tests for structured usability. We performed the tests with students and teachers at universities and schools in Denmark, Poland, and Portugal, where the trial experiments took place.

Our objective for the evaluation was to take the network technology to the user level to get qualitative results and evaluate the users' interactions with the applications. We wanted to know, for example, how users perceive the QoS enhancements, how they use them, and if they really understand the consequences of their choices.

We based the evaluation on three methods:[12-14]

Figure 6 shows the protocol stack's architecture.

To understand the complexity implied by the integrated services approach with IPv6 over ATM multicast, consider a simple scenario of an audioconference among three participants sending and receiving audio to a multicast IPv6 group. In this scenario, each client has to maintain 10 different ATM circuits to send and receive all application traffic classes: control traffic with MARS, unicast and multicast best-effort traffic, and multicast traffic with a guaranteed QoS. In addition, if the conference also includes video and it's sent to a different multicast address, the client has to maintain six more circuits.

We integrated the whole protocol stack implemented for client machines in the standard Winsock 2 network services architecture. Winsock 2 defines standard programming interfaces to access IPv6 and RSVP network services. These interfaces include all the functions needed by an application to send and receive IPv6 packets to unicast and multicast destinations and all the necessary functions to access RSVP functionality.

The applications that demand QoS support for their communications have to access both interfaces (see Figure 6), creating two different types of connection endpoints (sockets): One type sends an expert evaluation, performed by a group of evaluators with a deep knowledge in human–computer interaction (HCI) design and usability issues; think-aloud sessions, to observe how users work with the applications, to understand the reasons behind users' actions; and focus group discussions, dedicated to sum up users' impressions of the applications, the network features, and the evaluation.

Several interesting conclusions arose from the usability tests managed and performed by the Danish IT Center for Education and Research (UNI-C). In general, the results were positive. The users were able to use the QoS controls and understand their meaning and consequences of applications' behavior. Although some usability problems were found in the design of applications, the evaluations showed that even novice users in a relatively short time could learn to control the facilities offered by the applications and the underlying network.

Although users thought the interfaces of the DVL and Virtual Workspace were adequate, UNI-C made an important list of recommendations to improve them. However, the videoconference application interface was too complex, because it required a deep knowledge of the underlying technology to use it properly. UNI-C recommended redesigning the interface to make it usable for nontechnical users.

Most of the problems reported in this area could have been efficiently solved if the usability experts had participated in the development of applications. However, in our case that wasn't possible, since most of the applications used were already developed before BTI began.

We also obtained some other interesting conclusions from the evaluation phase:

- In general, application user interfaces play a dominant role in achieving satisfaction from end users and thus should be considered vital. This fact applies, of course, to how QoS enhancements are exposed in the user interface. Users have no idea about the underlying technical issues and aren't willing to modify their behavior to a network-oriented way of thinking. Therefore, none of the underlying complexities or details of the QoS model involved in the network should reach users. However, they should understand some basic network concepts like bandwidth.

- QoS control must be defined in such a way so that average users clearly understand it—for example, defining quality levels with simple terms like high, moderate, or low. Other controls, like continuous scale controls, present users with the question How much quality is enough? Application and network experts need to solve the issue of how to map the discrete values into traffic contracts or profiles.

Users must understand the consequences of QoS differentiation, relating the quality they get to the cost they pay, regardless of whether the service provider really applies any tariffs.

- Users must understand the consequences of QoS differentiation, relating the quality they get to the cost they pay, regardless of whether the service provider really applies any tariffs. Otherwise, users will always choose the high-quality profile, ignoring the consequence of their choice.

Conclusions and further work

The BTI project succeeded in adapting distant work and education applications to support QoS over an IPv6/ATM access network based on the integrated services model. However, the adaptation was in general hard to achieve. In the case of the Virtual Workspace environment, it required some important modifications to the application's communication model, to allow a shared reservation for all the tools included. These modifications were basically needed for lack of a simple mechanism to locally aggregate into a single QoS reservation the different traffic flows exchanged between client tools and server modules. Such an aggregation mechanism—that easily could have been based in flow identifiers found in IPv6 headers—would have greatly simplified the adaptation.

In addition, using such an aggregation mechanism could help reduce the number of flows that applications manage. For example, audio and video flows of a videoconference could be merged into one flow, thereby improving the network's scalability, which, as mentioned before, is compromised because of the high number of maintained ATM circuits. In addition, applications could easily handle local aggregation because they recognize the nature of flows and their similarity; the network can perform the same aggregation task but at a higher cost in terms of resources.

In the case of the DVL, the main problems arose from the fact that the original application's source code wasn't available. All the QoS-related functions and the adaptation to IPv6 had to be done using external proxy modules, which complicated the adaptation. In some cases, we had to investigate the original application's behavior, for example, to find out why the system was sensitive to packet loss experienced when the video bit rate exceeded the reserved resources.

Apart from the BTI project, the DVL has been extensively tested by Polish service providers. They've suggested some improvements to deploy the application as a commercial service. In particular, they suggested adding modules for session monitoring, billing, and access authorization. From the operator's point of view it's essential to examine the overall system performance (in the Asymmetric Digital Subscriber Line access environment), knowing the number of simultaneously delivered video streams.

The DVL system is currently under deployment at the Cracow Centre for Telemedicine to acquire video recordings of heart surgery to facilitate students' learning and support surgeons preparing themselves for operations.

Another improvement to the DVL, currently being worked out in another project, has to do with managing access to video episodes. At the moment, the system administrator manually fills in the database that allows direct access to video episodes. To make this task less time consuming, we're developing a module that automatically indexes content based on face, text, and voice recognition.

Finally, it's worth mentioning that the types of protocols we used in the project—most of them were prototypes and some of them didn't even reach beta status—had a big impact on the project's global objectives. Although adapting the applications to work in the BTI network and evaluating them with real users were at the beginning among the main project objectives, the protocol stack's development emerged as quite an unexpected and challenging task. Although we adapted the applications and successfully conducted usability tests, we feel that more stable protocol solutions would have allowed us to reach deeper conclusions concerning how QoS can be integrated and managed in distant work and education environments. **MM**

Acknowledgments

The work presented in this article was funded by the EU in the project Advanced Communications Technologies and Services (ACTS) 362 Broadband Trial Integration (BTI) and results from the project ACTS 109 Learn from Video Extensive Real ATM Gigabit (Leverage).

We wish to express our gratitude to Lars Blomgren Andersen and the members of the Danish IT Center for Education and Research (UNI-C) as well as to other anonymous evaluators for their work and effort during the usability evaluation of the BTI application setup.

References

1. N.E. Andersen et al., "Broadbandloop: A Full-Service Access Network for Residential and Small Business Users," *IEEE Comm. Mag.*, vol. 35, no. 12, Dec. 1997, pp. 88-93.
2. A. Azcorra, et al., "IP/ATM Integrated Services over Broadband Access Copper Technologies," *IEEE Comm. Mag.*, May 1999, pp. 90-97.
3. D.P. Hong and C. Albuquerque, "Evaluating the Impact of Emerging Streaming Media Applications on TCP/IP Performance," *IEEE Comm. Mag.*, vol. 39, no. 4, April 2001, pp. 76-82.
4. D. Fernández, L. Bellido, and E. Pastor, "Session Management and Collaboration in LEVERAGE," *Proc. First LEVERAGE Conf. Broadband Comm. Education and Training*, 1998, http://www.dit.upm.es/leverage.
5. P. Pacyna et al., "QoS-Aware Digital Video Retrieval Application," M.R. Syed, ed., *Multimedia Networking: Technology, Management and Applications*, Idea Group Publishing, Hershey, Penn., 2001.
6. J. Gozdecki, P. Pacyna, and Z. Papir, "Video Retrieval Service over an ATM Access Network under IPv6/RSVP Protocols," *J. Applied Systems Studies*, vol. 1, no. 2, 2000, pp. 237-253.
7. Z. Papir, R. Stankiewicz, and A. Szymanski, "Network-Based Digital Video Library System", *Proc. 10th Int'l Packet Video Workshop* (Packet Video 2000), CD-ROM, Dept. Electrical and Electronic Eng., Univ. of Cagliari, Sardinia, Italy, 2000.
8. F. Halsall, *Multimedia Communications: Applications, Networks, Protocols and Standards*, Addison Wesley, Reading, Mass., 2001.
9. N.E. Andersen et al., "Applying QoS Control through Integration of IP and ATM," *IEEE Comm.*, vol. 38, no. 7, July 2000, pp. 130-136.
10. Z. Wang, *Internet QoS: Architectures and Mechanisms for Quality of Service*, Morgan Kaufmann, San

Francisco, 2001.

11. D. Fernández et al., "Implementing the Integrated Services QoS Model with IPv6 over ATM Networks," *Proc. Network and Services for the Information Soc. 5th IFIP TC6 Int'l Symp.* (Interworking 2000), Lecture Notes in Computer Science, vol. 1938, Springer-Verlag, Berlin, 2000, pp. 83-95.
12. A.J. Dix et al., *Human-Computer Interaction*, Prentice-Hall, Upper Saddle River, N.J. 1997.
13. J. Nielsen, *Usability Engineering*, Morgan Kaufmann, San Francisco, 1994.
14. J.S. Dumas et al., *A Practical Guide to Usability Testing*, Ablex Publishing, Westport, Conn., 1994.

David Fernández is an associate professor of computer networks at the Telematic Systems Engineering Department of the Technical University of Madrid. He received his MSc degree in 1988 and his PhD in telematics engineering in 1993, both from the Technical University of Madrid. Since 1989, he has actively participated in several national and international research projects focused on formal description techniques, computer-supported cooperative work, and advanced Internet protocols. His current research interest includes multimedia protocols and QoS support for next generation Internet.

Ana B. Garcia received her MSc degree in telecommunications from the Technical University of Madrid in 1998. She is currently working on her PhD in computer science at the Telematic Systems Engineering Department from the Technical University of Madrid. Her research interests include the provisioning of QoS in Internet, advanced traffic analysis in IP backbones, and broadband IP networks.

David Larrabeiti is an associate professor of computer networking at the University Carlos III of Madrid. He obtained his PhD in computer science from the Technical University of Madrid in 1996. He has participated in international research projects related to the study and deployment of next generation networks and services. Recently, he has been involved in the development of multicast IP over ATM multipoint WAN networks and applications to support large international videoconferencing events. His current interests include developing multimedia protocol architectures, QoS, and active networking technology.

Arturo Azcorra is an associate professor of telematics engineering at the Technical University of Madrid (UPM, Spain. He received a BS in telecommunications engineering in 1986 and a PhD in computer science in 1989, both from UPM. His research projects have included protocol design, protocol engineering, advanced networks, and multimedia systems. He has published numerous papers in the field of advanced communications.

Piotr Pacyna is a lecturer on broadband network communications and operating systems in the Department of Telecommunications of the AGH University of Technology, Cracow, Poland, where he received an MS in computer sciences in 1995. He has spent sabbatical leaves working on traffic modeling for video flows and on ACTS projects. He's the coauthor of three books, several research papers, and expertise reports for the industry. His interests focus on broadband communications, video coding and transmission, and next-generation IP.

Zdzislaw Papir is a professor in the Department of Telecommunications at the AGH University of Technology in Cracow, Poland. He's currently lecturing on signal theory, modulation and detection theory, and modeling of telecommunication networks. He received his MSc in telecommunications in 1976 and his PhD in computer networks in 1979, both from the AGH University of Technology. His current research interests include performance analysis of digital modulations used in broadband access networks and integration of IP and ATM networking for provisioning multimedia services.

Readers may contact Papir at AGH University of Technology, Al. Mickiewicza 30, 30-059 Cracow, Poland, email papir@kt.agh.edu.pl.

For further information on this or any other computing topic, please visit our Digital Library at http://computer.org/publications/dlib.